SIEMENS

SIMATIC

S7 - OPEN MODBUS/TCP Kommunikation über die integrierte PN-Schnittstelle der CPU

Handbuch

SIEMENS

SIMATIC S7

S7 - OPEN MODBUS/TCP Kommunikation über die integrierte PN-Schnittstelle der CPU

Handbuch

Vorwort, Inhaltsverzeichnis

Produktbeschreibung 1

Getting Started 2

Inbetriebnahme 3

Parametrierung 4

Lizenzierung 5

FB MODBUSPN 6

Diagnose 7

Applikationsbeispiel 8

Anhänge

Literatur

Glossar

Ausgabe 2.1

Sicherheitstechnische Hinweise

Dieses Handbuch enthält Hinweise, die Sie zu Ihrer persönlichen Sicherheit, sowie zur Vermeidung von Sachschäden beachten müssen. Die Hinweise sind durch ein Warndreieck hervorgehoben und je nach Gefährdungsgrad folgendermaßen dargestellt:

Gefahr

bedeutet, dass Tod, schwere Körperverletzung oder erheblicher Sachschaden eintreten **werden**, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.

Warnung

bedeutet, dass Tod, schwere Körperverletzung oder erheblicher Sachschaden eintreten **können**, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.

Vorsicht

bedeutet, dass eine leichte Körperverletzung oder ein Sachschaden eintreten können, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.

Hinweis

ist eine wichtige Information über das Produkt, die Handhabung des Produktes oder den jeweiligen Teil der Dokumentation, auf den besonders aufmerksam gemacht werden soll.

Qualifiziertes Personal

Inbetriebsetzung und Betrieb eines Gerätes dürfen nur von **qualifiziertem Personal** vorgenommen werden. Qualifiziertes Personal im Sinne der sicherheitstechnischen Hinweise dieses Handbuchs sind Personen, welche die Berechtigung haben, Geräte, Systeme und Stromkreise gemäß den Standards der Sicherheitstechnik in Betrieb zu nehmen, zu erden und zu kennzeichnen.

Bestimmungsgemäßer Gebrauch

Beachten Sie folgendes:

Warnung

Das Gerät darf nur für die im Katalog und in der technischen Beschreibung vorgesehenen Einsatzfälle und nur in Verbindung mit von Siemens empfohlenen bzw. zugelassenen Fremdgeräten und -Komponenten verwendet werden.

Der einwandfreie und sichere Betrieb des Produktes setzt sachgemäßen Transport, sachgemäße Lagerung, Aufstellung und Montage sowie sorgfältige Bedienung und Instandhaltung voraus.

Marken

SIMATIC® und SIMATIC NET® sind eingetragenes Warenzeichen der SIEMENS AG.

Die übrigen Bezeichnungen in dieser Schrift können Marken sein, deren Benutzung durch Dritte für deren Zwecke die Rechte der Inhaber verletzen können.

Copyright © Siemens AG 2008 All Rights Reserved

Weitergabe sowie Vervielfältigung dieser Unterlage, Verwertung und Mitteilung ihres Inhalts ist nicht gestattet, soweit nicht ausdrücklich zugestanden. Zuwiderhandlungen verpflichten zu Schadensersatz. Alle Rechte vorbehalten, insbesondere für den Fall der Patenterteilung oder GM-Eintragung.

Siemens AG
Industry Sector
Engineering and Construction
I IS IN E&C
Postfach 3240, D- 91050 Erlangen
IT4.Industry@siemens.com

Haftungsausschluss

Wie haben den Inhalt der Druckschrift auf Übereinstimmung mit der beschriebenen Hard- und Software geprüft. Dennoch können Abweichungen nicht ausgeschlossen werden, so dass wir für die vollständige Übereinstimmung keine Gewähr übernehmen. Die Angaben in dieser Druckschrift werden regelmäßig überprüft, und notwendige Korrekturen sind in den nachfolgenden Auflagen enthalten. Für Verbesserungsvorschläge sind wir dankbar.

Technische Änderungen bleiben vorbehalten.

Vorwort

Zweck des Handbuchs

Die Informationen dieses Handbuchs ermöglichen es Ihnen, eine Kopplung zwischen einer CPU mit integrierter PN-Schnittstelle und einem Gerät, welches das Protokoll Open MODBUS/TCP unterstützt, aufzubauen und in Betrieb zu nehmen.

Inhalte des Handbuchs

Im vorliegenden Handbuch ist die Funktion des Modbus-Funktionsbausteins und dessen Parametrierung beschrieben.

Das Handbuch beinhaltet folgende Themen:

- Produktbeschreibung
- Getting Started
- Inbetriebnahme
- Parametrierung
- Lizenzierung
- Funktionsbaustein MODBUSPN
- Diagnose
- Applikationsbeispiel

Gültigkeitsbereich des Handbuchs

Das vorliegende Handbuch ist gültig für folgende Software:

Produkt	Identifizierungsnummer	ab Version
OPEN MODBUS/TCP	2XV9 450-1MB02	2.1
FB 102 „MODBUSPN“		3.1
FB 103 „TCP_COMM“		3.0
FB 104 „MOD_CLI“		1.1
FB 105 „MOD_SERV“		1.0

Hinweis

Das vorliegende Handbuch enthält die Beschreibung des FBs, wie sie zum Zeitpunkt der Herausgabe des Handbuchs gültig ist.

Weiterführende Informationsquellen

Alle weiteren Informationen bezüglich der PN-CPUen und der IM 151-8 PN/DP CPU (Montage, Inbetriebnahme etc.) entnehmen sie bitte dem Handbuch:

SIEMENS
SIMATIC S7-300
CPU 31xC und CPU 31x: Aufbauen
Betriebsanleitung
A5E00105491-07

SIEMENS
SIMATIC S7-300
CPU 31xC und CPU 31x, Technische Daten
Gerätehandbuch
A5E00105474-07

SIEMENS
SIMATIC S7-400
Automatisierungssystem S7-400: Aufbauen
Betriebsanleitung
A5E00850740-01

SIEMENS
SIMATIC S7-400
Automatisierungssystem S7-400 CPU-Daten
Gerätehandbuch
A5E00850745-06

SIEMENS
SIMATIC
Dezentrale Peripherie ET 200S
Interfacemodul IM151-8 PN/DP CPU
Betriebsanleitung
A5E02049033-01
SIEMENS

Produkt Information on
CPU315-2 PN/DP, 6ES7315-2EH13-0AB0
CPU315F-2 PN/DP, 6ES7315-2FH13-0AB0
CPU317-2 PN/DP, 6ES7317-2EK13-0AB0
CPU317F-2 PN/DP, 6ES7317-2FK13-0AB0
CPU317-2 DP, 6ES7317-2AJ10-0AB0
CPU317F-2 DP, 6ES7317-6FF03-0AB0
CPU319-3 PN/DP, 6ES7318-3EL00-0AB0
CPU319F-3 PN/DP, 6ES7318-3FL00-0AB0
A5E01103134-03

Weitere Informationen bezüglich STEP7 entnehmen Sie bitte den folgenden Handbüchern:

SIEMENS
SIMATIC Software
Basissoftware für S7 und M7
STEP7 Benutzerhandbuch
C79000-G7000-C502-..

SIEMENS
SIMATIC Software
Systemsoftware für S7-300/400
System- und Standardfunktionen
Referenzhandbuch
C79000-G7000-C503-02

Rückfragen

Bei Fragen zur Nutzung der in diesem Handbuch beschriebenen FBs, die Sie hier nicht beantwortet finden, wenden Sie sich bitte an Ihren Siemens-Ansprechpartner, von dem Sie diesen Funktionsbaustein erhalten haben.

Konventionen

In der vorliegenden Dokumentation wird im Folgenden die Bezeichnung PN-CPU verwendet. Die Ausführungen sind für die **PN-CPU**en der Reihe **315, 317, 319, 414** und **416** sowie für die **IM 151-8 PN/DP CPU** gültig.

Einsatzbereich

Die in diesem Handbuch beschriebenen Funktionsbausteine stellen eine Verbindung zwischen einer PN-CPU und Modbusgeräten anderer Hersteller dar.

Inhaltsverzeichnis

1 Produktbeschreibung 1-1

1.1 Einsatzmöglichkeiten 1-1

1.2 Hard- und Softwarevoraussetzungen 1-1

2 Getting Started 2-1

3 Inbetriebnahme 3-1

3.1 Installieren der Bibliothek auf dem STEP7-PG/-PC 3-1

3.2 CPU – IP-Adresse zuweisen 3-2

3.3 Einfügen der Funktionsbausteine 3-4

4 Parametrieren der Modbus-Kommunikation 4-1

4.1 Parametrieren mit dem Wizard 4-2

4.2 Manuelle Parametrierung 4-3

5 Lizenzierung 5-1

6 Funktionsbaustein MODBUSPN 6-1

6.1 Funktionsweise der FB 6-1

6.2 Parameter des Funktionsbausteins MODBUSPN 6-5

6.3 Adressabbildung 6-12

6.4 Vom FB verwendete Daten und Standardfunktionen 6-15

6.5 Umbenennen von Standardfunktionen 6-16

7 Diagnose 7-1

7.1 Diagnose über die Anzeigeelemente der CPU 7-2

7.2 Überprüfungen durch den FB MODBUSPN 7-3

7.3 Diagnosemeldungen des FB MODBUSPN 7-6

7.4 Diagnosemeldungen der verwendeten Bausteine 7-12

7.5 Diagnosemeldungen des SFC24 7-12

8 Applikationsbeispiel 8-1

A Literatur 1

1 Produktbeschreibung

1.1 Einsatzmöglichkeiten

Einordnen in die Systemumgebung

Der vorliegende Funktionsbaustein stellt ein Software-Produkt für CPUs mit integrierter PN-Schnittstelle der SIMATIC S7-300, S7-400 und IM 151-8 PN/DP CPU dar.

Funktion der FBs

Mit diesen Funktionsbausteinen wird eine Kommunikation zwischen einer SIMATIC-CPU mit integrierter PN-Schnittstelle und einem Gerät, welches das Protokoll OPEN MODBUS/TCP unterstützt, ermöglicht.

Die Datenübertragung wird nach dem Client-Server-Prinzip abgewickelt.

Die SIMATIC S7 kann bei der Übertragung sowohl als Client als auch als Server betrieben werden.

Verwendung der Portnummer 502

Das Protokoll MODBUS/TCP läuft üblicherweise über den Port 502. Diese Portnummer ist noch nicht für alle PN-CPUen freigegeben und derzeit nur bei Verwendung einer CPU 319, IM 151-8 PN/DP CPU, CPU 414 oder CPU 416 mit entsprechender Firmwareversion möglich. Die Information bezüglich der Freigabe finden Sie im jeweiligen Handbuch Ihrer CPU.

Die CPU lässt nur eine **einmalige Verwendung einer bestimmten Port-Nummer** zu. Die CPU kann also nicht gleichzeitig von mehreren Geräten auf derselben Port-Nummer angesprochen werden.

1.2 Hard- und Softwarevoraussetzungen

Verwendbare Baugruppen für MODBUSPN

Die Funktionsbausteine wurden auf den CPUen mit der MLFB

CPU315-2 PN/DP	6ES7315-2EG10-0AB0
CPU317-2 PN/DP	6ES7317-2EK13-0AB0
CPU319-3 PN/DP	6ES7318-3EL00-0AB0
CPU414-3 PN/DP	6ES7414-3EM05-0AB0
CPU416-3 PN/DP	6ES7416-3FR05-0AB0
IM151-8 PN/DP	6ES7151-8AB00-0AB0

getestet.

Die aktuellen Hardwarevoraussetzungen entnehmen Sie bitte hier: www.siemens.com/s7modbus.

Softwareausgabe-stände

Der Einsatz des FBs MODBUSPN ist ab **STEP7-Version 5.4 SP4** möglich.

Speicherbedarf

Der FB MODBUSPN benötigt 6048 Byte Arbeitsspeicher und 7294 Byte Ladespeicher.

Der FB MOD_CLI benötigt 11184 Byte Arbeitsspeicher und 12054 Byte Ladespeicher.

Der FB MOD_SERV benötigt 10660 Byte Arbeitsspeicher und 11452 Byte Ladespeicher.

Der FB TCP_COMM benötigt 1880 Byte Arbeitsspeicher und 2224 Byte Ladespeicher.

2 Getting Started

Vorgehensweise

1. Installation von „OpenModbusTCP PN CPU“ und Einfügen der Modbusbausteine in das Anwenderprojekt
   => Kapitel 3.1 bis 3.3

2. Parametrierung der Verbindungsparameter entsprechend der Anlagenkonstellation (IP-Adresse, Portnummer etc.)
   => Kapitel 4.1 und 4.2

3. Parametrierung des Parameter-DBs MODBUS_PARAM entsprechend den Anforderungen (Client/Server, Verbindungsaufbau bei Neustart, Modbusregister, DB-Bereiche etc.)
   => Kapitel 4.1 und 4.2

4. Aufruf des Modbusbausteins FB102 in den notwendigen OBs
   => Kapitel 6.1

5. Parametrierung des Modbusbausteins für die Initialisierung und für die Laufzeit
   => Kapitel 6.2

6. Laden des Anwenderprogramms in die CPU und Lizenzierung des Modbusbausteins für diese CPU
   => Kapitel 5

3 Inbetriebnahme

Allgemeines Die im Folgenden verwendeten Angaben zu STEP7 beziehen sich auf die STEP7-Version 5.4 SP4. Bei späteren Versionen können Abläufe, Namens- und Verzeichnisangaben geändert sein.

Voraussetzungen STEP7–Grundkenntnisse, AWL–Kenntnisse, SPS–Grundkenntnisse

3.1 Installieren der Bibliothek auf dem STEP7-PG/-PC

Lieferumfang Die beiliegende CD enthält ein Setup, mit dem die Bibliothek „**Modbus_PN_CPU**“, das Beispielprojekt und die Handbücher in Deutsch und Englisch in den entsprechenden STEP7-Verzeichnissen installiert werden.

Zusätzlich befinden sich auf der CD die Handbücher im PDF – Format.

Voraussetzungen Um die Installation durchführen zu können, muss vorher die Projektierungssoftware **STEP7** installiert worden sein.

Installation Legen Sie die Modbus-CD in das CD-ROM-Laufwerk Ihres PGs/PCs ein. Wenn das Setupprogramm nicht automatisch startet, erfolgt die Installation folgendermaßen:

1. Wählen Sie im Windows Explorer das CD-ROM-Laufwerk, öffnen Sie das Verzeichnis Setup und starten Sie die Datei **Setup.exe**.
2. Befolgen Sie Schritt für Schritt die Anweisungen, die Ihnen das Installationsprogramm anzeigt.

Sie finden nun

- die Bibliothek in \Program Files\Siemens\Step7\S7libs,
- das Beispielprojekt in \Program Files\Siemens\Step7\Examples,
- das Handbuch in \Program Files\Siemens\Step7\S7manual\S7Comm,
- das Software Registration Form in \Program Files\Siemens\Step7\S7libs\Modbus_PN_CPU.

Beim ersten Aufruf der Modbus-Bibliothek verwenden Sie bitte die Funktion „Durchsuchen“ des Öffnen-Dialogs um auf die Bibliothek zuzugreifen.

Das Handbuch kann auch über den Shortcut unter \Program Files\Siemens \Dokumentation geöffnet werden.

3.2 CPU – IP-Adresse zuweisen

Einleitung

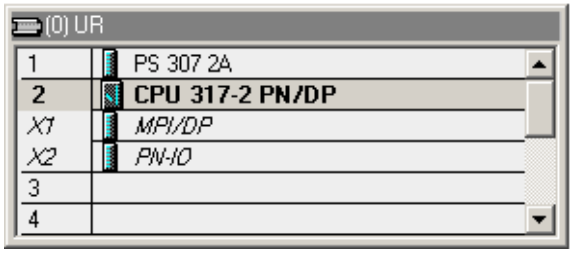
Jeder Teilnehmer eines Ethernet-Netzwerks ist über eine weltweit eindeutige Adresse identifizierbar. Diese sogenannte MAC-Adresse wird vom Hersteller vorgegeben und kann vom Anwender nicht verändert werden.

In den folgenden Schritten werden Sie dieser physikalischen Adresse eine IP-Adresse im Ethernet zuweisen.
In diesem Beispiel wird eine CPU 317-2 PN/DP eingefügt.

Vorgehensweise

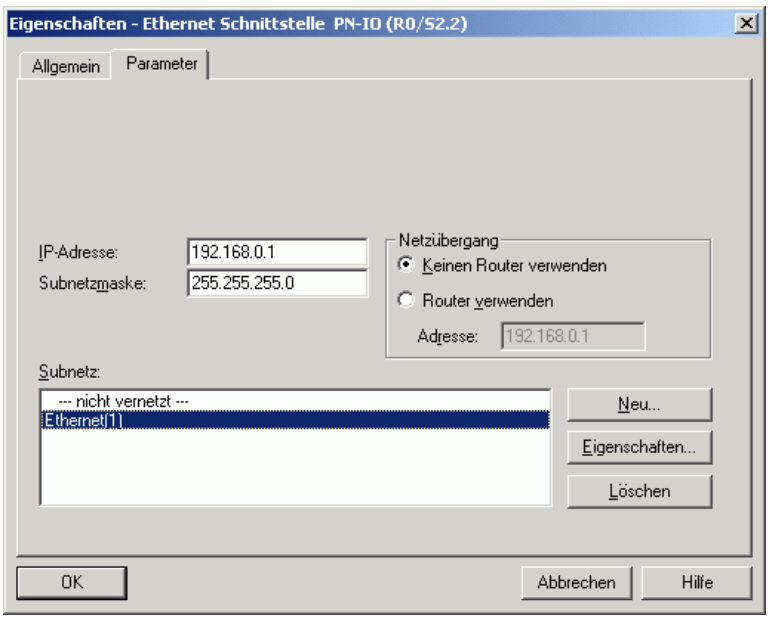
Bevor Sie die Konfiguration vornehmen können, müssen Sie mit STEP7 ein **S7-Projekt** angelegt haben.

1. Öffnen Sie HWKonfig.
   Fügen Sie auf Steckplatz 2 eine CPU 317-2 PN/DP ein. Das Eigenschaftsfenster der PN-IO-Schnittstelle X2 wird angezeigt.



2. Durch Doppelklick auf die Zeile X2 öffnet sich der Eigenschaftsdialog der PN-IO-Schnittstelle

3. Geben Sie die IP-Adresse und die Subnetzmaske ein.
   Wenn Sie eine Verbindung über einen Router aufbauen, müssen Sie zusätzlich noch die Adresse des Routers eingeben.



4. Klicken Sie auf die Schaltfläche "Neu..." und vergeben Sie einen Namen für ein neues Industrial Ethernet Subnetz. Klicken Sie dann auf die Schaltfläche "OK".
   Ergebnis: Sie haben ein neues Industrial Ethernet-Subnetz angelegt.

5. Klicken Sie auf die Schaltfläche "OK".
   Ergebnis: Das Eigenschaftsfenster der PN-IO-Schnittstelle der CPU 317-2 PN/DP wird geschlossen.

3.3 Einfügen der Funktionsbausteine

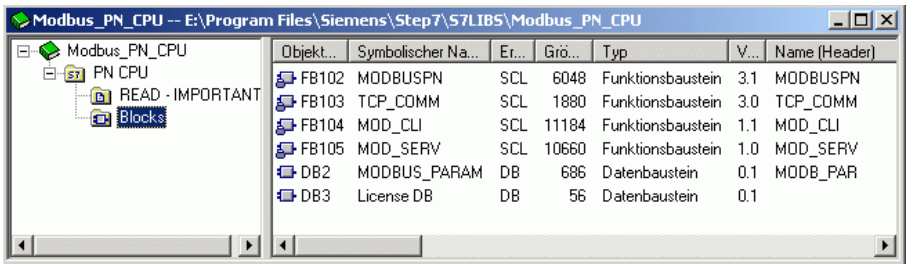
Bausteine der Modbus-Bibliothek

Folgende FBs werden für die Modbuskommunikation benötigt:

- MODBUSPN
- MOD_CLI
- MOD_SERV
- TCP_COMM

Um diese in Ihr Projekt einfügen zu können, müssen Sie die Bausteine aus der Bibliothek kopieren.

Des Weiteren befindet sich der Parameterdatenbaustein MODBUS_PARAM als Vorlage in der Bibliothek. Diese können Sie zur Arbeitserleichterung ebenfalls in Ihr Projekt kopieren.


Modbus_PN_CPU -- E:\Program Files\Siemens\Step7\S7LIBS\Modbus_PN_CPU
Modbus_PN_CPU
PN CPU
READ - IMPORTANT
Blocks
Objekt... Symbolischer Na... Er... Grö... Typ V... Name (Header)
FB102 MODBUSPN SCL 6048 Funktionsbaustein 3.1 MODBUSPN
FB103 TCP_COMM SCL 1880 Funktionsbaustein 3.0 TCP_COMM
FB104 MOD_CLI SCL 11184 Funktionsbaustein 1.1 MOD_CLI
FB105 MOD_SERV SCL 10660 Funktionsbaustein 1.0 MOD_SERV
DB2 MODBUS_PARAM DB 686 Datenbaustein 0.1 MODB_PAR
DB3 License DB DB 56 Datenbaustein 0.1

Bausteine der Standard Library

Folgende FBs werden für die Modbuskommunikation benötigt:

- TSEND (FB63)
- TRCV (FB64)
- TCON (FB65)
- TDISCON (FB66).

Diese Kommunikationsbausteine finden Sie in der Bibliothek „**Standard Library → Communication Blocks**“ und müssen auch in Ihr Projekt eingefügt werden.

Beachten Sie, dass folgende Versionen der FBs Voraussetzung für den einwandfreien Betrieb des FB MODBUSPN sind:

TSEND	V2.1 oder höher
TRCV	V2.2 oder höher
TCON	V2.3 oder höher
TDISCON	V2.1 oder höher

Weiterhin wird der Baustein FC10 EQ_STRNG aus der Bibliothek „**Standard Library → IEC Function Blocks**“ benötigt.

4 Parametrieren der Modbus-Kommunikation

Allgemeines

Für die Kommunikation über die integrierte PN-Schnittstelle der CPU ist keine Verbindungsprojektierung in NetPro notwendig. Die Verbindungen werden mit Hilfe der Funktionsbausteine TCON (FB65) und TDISCON (FB66) auf- bzw. abgebaut.

Es können mehrere Verbindungen zu verschiedenen Kommunikationspartnern projektiert und gleichzeitig aufgebaut werden. Die Anzahl der gleichzeitig aufgebauten Verbindungen ist CPU-abhängig.

Verbindungs-Datenbaustein MODBUS_PARAM

Die für den Aufbau der Verbindungen und die Bearbeitung der Modbustelegramme notwendigen Daten werden in einer Struktur in einem Datenbaustein – dem Parameterdatenbaustein MODBUS_PARAM –definiert. Dabei werden zuerst die verbindungsspezifischen Daten abgelegt. Nach den verbindungsspezifischen Daten folgen die Modbusparameter.

Für jede Verbindung wird **1 Struktur** benötigt, in dem die Verbindungsparameter zwischen den Kommunikationspartnern und die Modbusparameter definiert werden. Für jede weitere Verbindung kann der Datenbaustein erweitert oder ein neuer angelegt werden und die passende Struktur mit den Verbindungs- und den Modbusparametern eingefügt werden.

In dem Parameterdatenbaustein können die Projektierungsdaten aller Verbindungen enthalten sein. Es ist ebenso möglich, für jede Verbindung einen separaten Parameterdatenbaustein anzulegen.

Eine vorgefertigte Struktur ist als Beispiel in der Bibliothek „Modbus_PN_CPU“ enthalten.

Aufbau des DB MODBUS_PARAM:

Adresse	Name
	STRUCT
+0.0	Verbindung 1: Verbindungsparameter
+64.0	Verbindung 1: Modbusparameter
	END_STRUCT
	STRUCT
+650.0	Verbindung 2: Verbindungsparameter
+714.0	Verbindung 2: Modbusparameter
	END_STRUCT
...	**...**
	STRUCT
650*i	Verbindung i+1: Verbindungsparameter (65+i)
650*i+64	Verbindung i+1: Modbusparameter
	END_STRUCT

Verbindungs-parameter

In dem 1. Block werden die verbindungsspezifischen Parameter, wie z.B. die lokal verwendete Hardwareschnittstelle und die IP-Adresse des Kommunikationspartners, definiert. Mit Hilfe dieser Parameter können die Funktionen TCON und TDISCON eine Verbindung auf- bzw. abbauen. Die genaue Struktur finden Sie in Kapitel 4.2.

Die Datenstruktur des Verbindungsparameterblocks muss zwingend eingehalten werden, da sonst keine Verbindung aufgebaut werden kann.

Modbusparameter

In den Modbusparametern werden die für die Betriebsart und Adressreferenz notwendigen Daten abgelegt, wie z.B. die Modbusbereiche, die in den Datenbausteinen abgebildet werden, oder die Betriebsart der S7 als Modbus Server oder Modbus Client. Die Datenstruktur der Modbusparameter muss eingehalten werden, da sonst keine fehlerfreie Bearbeitung möglich ist.

Projektierungs-möglichkeiten

Es gibt zwei Möglichkeiten, die Projektierung für die Verbindungs- und Modbusparameter vorzunehmen. Zum Einen ist die Eingabe über einen Wizard möglich, mit dem sehr komfortabel die Parametrierung vorgenommen werden kann. Zum Anderen können die Parameter durch Editieren der Struktur im Parameterdatenbaustein eingestellt werden.

Diese beiden Möglichkeiten werden in den folgenden Kapiteln 4.1 und 4.2 beschrieben.

4.1 Parametrieren mit dem Wizard

Allgemeines

Mit dem „**ModbusTCP Wizard**“ ist eine komfortable Projektierung der Verbindungen und der Modbusparameter im Parameterdatenbaustein MODBUS_PARAM möglich. Dabei wird die komplette Struktur (Verbindungsparameter und Modbusparameter) angelegt.

Es wird empfohlen für die Parametrierung des MODBUS_PARAM den Wizard einzusetzen.

Den Wizard finden Sie unter
http://support.automation.siemens.com/WW/view/de/31535566.

4.2 Manuelle Parametrierung

Vorgehensweise

Kopieren Sie dazu den DB2 aus der Bibliothek „**Modbus_PN_CPU**“ und fügen Sie diesen in ihr Projekt ein. Werden die Nummern bereits anderweitig verwendet, kann der DB umbenannt werden.

Die Parameter im Baustein MODBUS_PARAM dürfen während der Laufzeit nicht verändert werden. Nach einer Änderung der Parameter muss die CPU mit STOP -> RUN neu gestartet werden.

Aufbau und Anpassungen der Verbindungs-parameter

Pro Verbindung wird eine Struktur benötigt.

Adresse	Name	Typ	Anfangswert
0.0		STRUCT	
+0.0	OUCW_1	STRUCT	
+0.0	block_length	WORD	W#16#40
+2.0	id	WORD	W#16#1
+4.0	connection_type	BYTE	B#16#1
+5.0	active_est	BOOL	FALSE
+6.0	local_device_id	BYTE	B#16#2
+7.0	local_tsap_id_len	BYTE	B#16#2
+8.0	rem_subnet_id_len	BYTE	B#16#0
+9.0	rem_staddr_len	BYTE	B#16#0
+10.0	rem_tsap_id_len	BYTE	B#16#0
+11.0	next_staddr_len	BYTE	B#16#0
+12.0	local_tsap_id	ARRAY[1..16]	B#16#D0, B#16#7, B#16#0, B#16#0
*1.0		BYTE	
+28.0	rem_subnet_id	ARRAY[1..6]	B#16#0, B#16#0, B#16#0, B#16#0,
*1.0		BYTE	
+34.0	rem_staddr	ARRAY[1..6]	B#16#0, B#16#0, B#16#0, B#16#0,
*1.0		BYTE	
+40.0	rem_tsap_id	ARRAY[1..16]	B#16#0, B#16#0, B#16#0, B#16#0,
*1.0		BYTE	
+56.0	next_staddr	ARRAY[1..6]	B#16#0, B#16#0, B#16#0, B#16#0,
*1.0		BYTE	
+62.0	spare	WORD	W#16#0

block_length

Dieser Parameter bezeichnet die Länge der Verbindungsparameter und darf nicht verändert werden.

Fester Wert: W#16#40

id

Für jede logische Verbindung wird eine Verbindungs-ID vergeben. Diese muss im gesamten Parameterdatenbaustein eindeutig sein. Die ID wird beim Aufruf des FB MODBUSPN angegeben und bei den internen Aufrufen der T-Bausteine (TCON, TSEND, TRCV und TDISCON) verwendet.

Wertebereich: W#16#1 bis W#16#FFF

connection_type Hier wird der Verbindungstyp für den Aufbau der Verbindung durch die Funktion TCON definiert. Der einzustellende Wert ist CPU-abhängig.

TCP (Kompatibilitätsmode):	B#16#01 für CPU 315 bzw. 317 <= FW V2.3
TCP:	B#16#11 für CPU 315 bzw. 317 >= FW V2.4, IM 151-8 PN/DP CPU, CPU 319, CPU 414 und CPU 416

Je nach verwendeter Firmware können diese Angaben variieren.
Weitere Angaben finden Sie unter:
http://support.automation.siemens.com/WW/view/de/24294554

active_est Dieser Parameter bezeichnet die Art des Verbindungsaufbaus, aktiv oder passiv. Es wird empfohlen, dass der Modbus Client den aktiven und der Modbus Server den passiven Verbindungsaufbau übernimmt.

Aktiver Verbindungsaufbau:	TRUE
Passiver Verbindungsaufbau:	FALSE

local_device_id Mit der *local_device_id* wird die IE-Schnittstelle der verwendeten PN-CPU definiert. Je nach PN-CPU-Typ werden unterschiedliche Einstellungen benötigt.

IM 151-8 PN/DP CPU:	B#16#1
CPU 315 bzw. 317:	B#16#2
CPU 319:	B#16#3
CPU 414 bzw. CPU 416	B#16#5

local_tsap_id_len Es wird die Länge des Parameters *local_tsap_id* (= lokale Portnummer) angegeben.

Aktiver Verbindungsaufbau:	0
Passiver Verbindungsaufbau:	2

rem_subnet_id_len Dieser Parameter wird derzeit nicht verwendet und muss mit B#16#0 belegt sein.

rem_staddr_len Es wird die Länge des Parameters *rem-staddr*, also der IP-Adresse des Kommunikationspartners, angegeben. Soll über eine unspezifizierte Verbindung kommuniziert werden, wird keine IP-Adresse für den Partner angegebenen.

Unspezifizierte Verbindung:	B#16#0
Spezifizierte Verbindung:	B#16#4

rem_tsap_id_len Dieser Parameter bezeichnet die Länge des Parameters *rem_tsap_id*, der Portnummer des remoten Kommunikationspartners.

Aktiver Verbindungsaufbau:	2
Passiver Verbindungsaufbau:	0

next_staddr_len Hier wird die Länge des Parameters *next_staddr* festgelegt.

Bei PN-Schnittstelle: B#16#0

local_tsap_id Mit diesem Parameter wird die lokale Portnummer eingestellt. Die Art der Darstellung wird dabei abhängig des Parameters *connection_type* unterschieden. Der Wertebereich ist CPU-abhängig. Die Portnummer muss auf der CPU eindeutig sein.

Bei *connection_type* B#16#01

local_tsap_id[1]	low byte der Port-Nr. in Hex-Darstellung
local_tsap_id[2]	high byte der Port-Nr. in Hex-Darstellung
local_tsap_id[3-16]	B#16#00

Bei *connection_type* B#16#11

local_tsap_id[1]	high byte der Port-Nr. in Hex-Darstellung
local_tsap_id[2]	low byte der Port-Nr. in Hex-Darstellung
local_tsap_id[3-16]	B#16#00

rem_subnet_id Dieser Parameter wird derzeit nicht verwendet und muss mit 0 belegt werden.

rem_staddr In diesem Byte-Array wird die IP-Adresse des remoten Kommunikationspartners eingetragen. Im Falle einer unspezifizierten Verbindung wird keine IP-Adresse eingetragen. Die Art der Darstellung wird abhängig des Parameters *connection_type* unterschieden.

Beispiel: IP-Adresse 192.168.0.1:

Bei *connection_type* B#16#01

rem_staddr[1] =	B#16#01 (1),
rem_staddr[2] =	B#16#00 (0),
rem_staddr[3] =	B#16#A8 (168),
rem_staddr[4] =	B#16#C0 (192),
rem_staddr[5-6]=	B#16#00 (reserviert)

Bei *connection_type* B#16#11

rem_staddr[1] =	B#16#C0 (192),
rem_staddr[2] =	B#16#A8 (168),
rem_staddr[3] =	B#16#00 (0),
rem_staddr[4] =	B#16#01 (1),
rem_staddr[5-6]=	B#16#00 (reserviert)

rem_tsap_id Mit diesem Parameter wird die remote Portnummer eingestellt. Die Art der Darstellung wird dabei abhängig des Parameters *connection_type* unterschieden. Der Wertebereich ist CPU-abhängig.

Bei *connection_type* B#16#01

local_tsap_id[1]	low byte der Port-Nr. in Hex-Darstellung
local_tsap_id[2]	high byte der Port-Nr. in Hex-Darstellung
local_tsap_id[3-16]	B#16#00

Bei *connection_type* B#16#11

local_tsap_id[1]	high byte der Port-Nr. in Hex-Darstellung
local_tsap_id[2]	low byte der Port-Nr. in Hex-Darstellung
local_tsap_id[3-16]	B#16#00

next_staddr Dieser Parameter bezeichnet die Rack- und Steckplatznummer des verwendeten CPs. Bei Verwendung der integrierten PN-Schnittstelle der CPU muss dieser Parameter auf 0 gesetzt werden.

next_staddr[1-6]	B#16#00

spare Dieser Parameter wird nicht verwendet und muss mit 0 vorbelegt werden.

Anpassungen der Modbusparameter

Mit den Modbusparametern im Baustein MODBUS_PARAM werden die Betriebsart der Modbuskommunikation und die Adressabbildung von Modbusadressen auf SIMATIC-Adressen festgelegt.

+64.0	server_client	BOOL	TRUE
+64.1	single_write	BOOL	FALSE
+64.2	connect_at_startup	BOOL	FALSE
+65.0	reserved	BYTE	B#16#0
+66.0	data_type_1	BYTE	B#16#3
+68.0	db_1	WORD	W#16#B
+70.0	start_1	WORD	W#16#1
+72.0	end_1	WORD	W#16#1F4
+74.0	data_type_2	BYTE	B#16#3
+76.0	db_2	WORD	W#16#C
+78.0	start_2	WORD	W#16#2D0
+80.0	end_2	WORD	W#16#384
+82.0	data_type_3	BYTE	B#16#4
+84.0	db_3	WORD	W#16#D
+86.0	start_3	WORD	W#16#2D0
+88.0	end_3	WORD	W#16#3E8
+90.0	data_type_4	BYTE	B#16#0
+92.0	db_4	WORD	W#16#4
+94.0	start_4	WORD	W#16#0
+96.0	end_4	WORD	W#16#64
+98.0	data_type_5	BYTE	B#16#1
+100.0	db_5	WORD	W#16#E
+102.0	start_5	WORD	W#16#280
+104.0	end_5	WORD	W#16#4E2
+106.0	data_type_6	BYTE	B#16#2
+108.0	db_6	WORD	W#16#F
+110.0	start_6	WORD	W#16#6A4
+112.0	end_6	WORD	W#16#8FC
+114.0	data_type_7	BYTE	B#16#1
+116.0	db_7	WORD	W#16#10
+118.0	start_7	WORD	W#16#6A4
+120.0	end_7	WORD	W#16#8FC
+122.0	data_type_8	BYTE	B#16#0
+124.0	db_8	WORD	W#16#8
+126.0	start_8	WORD	W#16#0
+128.0	end_8	WORD	W#16#64
+130.0	internal_send_buffer	ARRAY[1..260]	B#16#0
*1.0		BYTE	
+390.0	internal_recv_buffer	ARRAY[1..260]	B#16#0
*1.0		BYTE	
=650.0		END_STRUCT	
=650.0		END_STRUCT	

server_client

TRUE: S7 ist Server
FALSE: S7 ist Client

single_write

In der Betriebsart „S7 ist Client“ wird mit dem Parameter single_write = TRUE bei schreibenden Aufträgen mit Länge 1 die Funktionscodes 5 und 6 verwendet.

Ist single_write = FALSE, werden bei allen schreibenden Aufträgen die Funktionscodes 15 und 16 verwendet.

connect_at_ startup

Hiermit wird der Zeitpunkt des Verbindungsaufbaus festgelegt.
Ist *connect_at_startup* auf TRUE gesetzt, darf erst ein Datenauftrag abgesetzt werden, wenn die Verbindung korrekt aufgebaut werden konnte (CONN_ESTABLISHED = TRUE) oder ein entsprechender Fehler an ERROR und STATUS_CONN angezeigt wird.

FALSE: Verbindungsaufbau bei gesetztem ENQ_ENR
TRUE: Verbindungsaufbau direkt nach Neustart

8 Datenbereiche

Es werden 8 Datenbereiche für die Abbildung der MODBUS Adressen im S7-Speicher angeboten. Es muss mindestens der erste Datenbereich definiert werden, die anderen 7 Datenbereiche sind optional. Aus den Datenbereichen wird abhängig vom Auftragstyp gelesen oder in sie geschrieben.

Mit einem Auftrag kann immer nur aus einem DB gelesen / in einen DB geschrieben werden. Zugriffe auf Register oder Bitwerte, die in mehreren DBs liegen, auch wenn die Nummern lückenlos hintereinander liegen, sind auf zwei Aufträge aufzuteilen. Dies ist bei der Projektierung zu berücksichtigen.

Es ist möglich in einen Datenbaustein mehr Modbusbereiche (Register oder Bitwerte) abzubilden als mit einem Telegramm bearbeitet werden können.

data_type_x

Mit dem Parameter *data_type_x* wird angegeben, welche MODBUS Datentypen in diesem Datenbaustein abgebildet werden.
Wird in *data_type_x* der Wert 0 eingetragen, wird der entsprechende Datenbereich nicht verwendet.

Kennung	Datentyp	Datenbreite
0	Bereich nicht verwendet	
1	Coils	Bit
2	Inputs	Bit
3	Holding Register	Word
4	Input Register	Word

db_x

Der Parameter *db_x* legt den Datenbaustein fest, in dem die nachfolgend definierten MODBUS Register oder Bitwerte abgebildet werden. Die DB-Nummer 0 ist nicht erlaubt, da diese für das System reserviert ist.

db_x
DB-Nummer 1 bis 65535 (W#16#0001 bis W#16#FFFF)

start_x
end_x

Mit *start_x* wird die erste Modbusadresse, die im Datenwort 0 des DB abgebildet wird, angegeben. Der Parameter *end_x* definiert die Adresse der letzten MODBUS-Adresse.

Bei Registerzugriffen berechnet sich die Datenwortnummer im S7 DB, in die die letzte Modbusadresse eingetragen wird, nach folgender Formel:

DBW Nummer = (*end_x* – *start_x*) ∗ 2

Bei Bitzugriffen berechnet sich die Datenbytenummer im S7 DB, in die die letzte Modbusadresse eingetragen wird, nach folgender Formel:

DBB Nummer = (*end_x* – *start_x* + 7) / 8

Die definierten Datenbereiche dürfen sich nicht überlappen. Der Parameter *end_x* darf nicht kleiner als *start_x* sein. Im Fehlerfall wird der Anlauf des FBs mit Fehler beendet. Sind beide Werte gleich, wird 1 Modbusadresse (1 Register oder 1 Bitwert) zugeordnet.

Im Kapitel 6.3 ist ein Beispiel für die Abbildung der MODBUS Adressen auf S7-Speicherbereiche dargestellt.

start_x, end_x

MODBUS Adresse	0 bis 65535 (W#16#0000 bis W#16#FFFF)

internal_send_ buffer

Dieses Array wird intern im FB für die Sendedaten verwendet. Zugriffe oder Änderungen in diesem Bereich sind nicht zulässig.

internal_recv_ buffer

Dieses Array wird intern im FB für die Empfangsdaten verwendet. Zugriffe oder Änderungen in diesem Bereich sind nicht zulässig.

5 Lizenzierung

Allgemeines

Der Baustein MODBUSPN muss auf jeder CPU einzeln lizenziert werden. Die Lizenzierung erfolgt in 2 Schritten: dem Auslesen des IDENT_CODES und der Eingabe des Freischaltcodes REG_KEY.

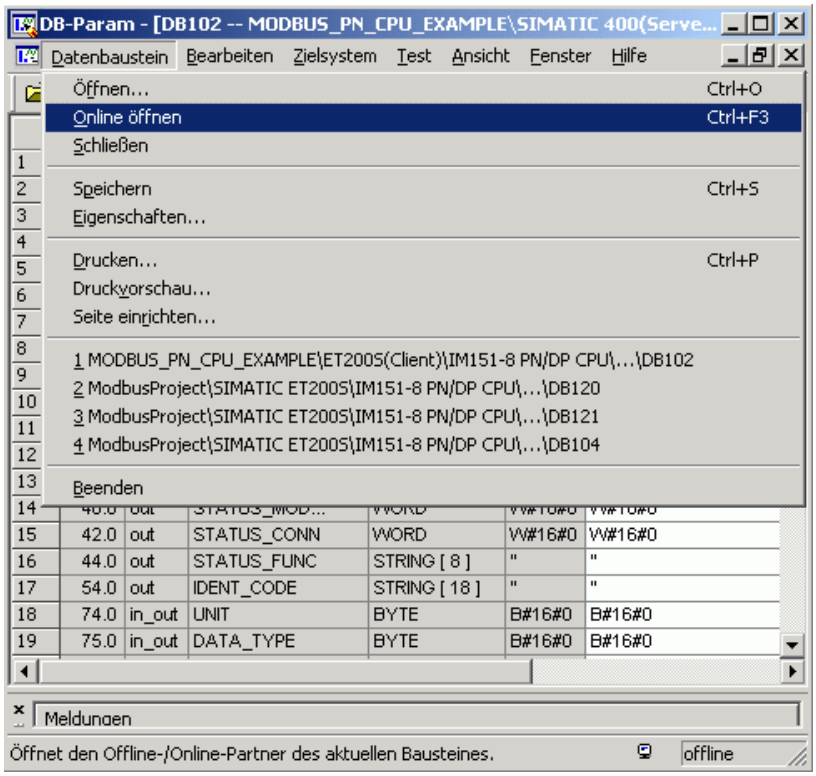
Auslesen des IDENT_CODES

Für das Auslesen des IDENT_CODES gehen Sie wie folgt vor:

1. Parametrieren Sie den Baustein MODBUSPN Ihren Anforderungen entsprechend in einem zyklischen OB, im OB100 und im OB121. Laden Sie das Programm in die CPU und setzen Sie diese in RUN.

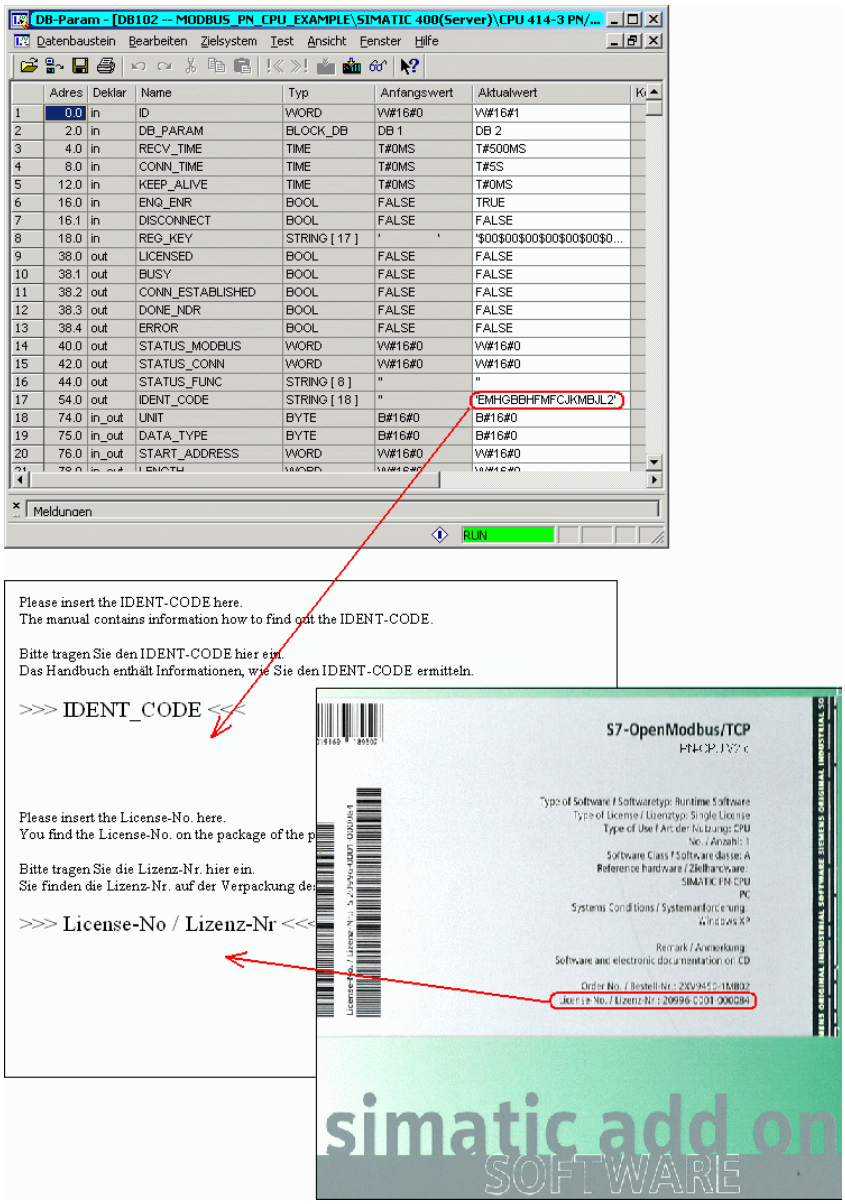
2. Öffnen Sie den Instanz-DB des Modbusbausteins. Gehen Sie über den Menüpunkt **„Datenbaustein“ -> „Online öffnen“** um den DB online zu öffnen.

   Ein ‚Baustein beobachten’ über den Button ist nicht ausreichend.



3. Am Ausgang IDENT_CODE wird eine 18stellige Zeichenfolge angezeigt.

   Kopieren Sie diesen String per Copy/Paste aus dem DB und fügen ihn in das Formular ***IT4Industry SOFTWARE REGISTRATION FORM*** ein. Dieses Formular wird bei der Installation im Bibliotheks-Pfad ..\Program Files\Siemens\Step7\S7LIBS\Modbus_PN_CPU abgelegt und liegt zusätzlich auch auf der Installations-CD. Tragen Sie die Lizenz-Nr. von der Produktverpackung in das Formular ein.


DB-Param - [DB102 -- MODBUS_PN_CPU_EXAMPLE\SIMATIC 400(Server)\CPU 414-3 PN/...
Datenbaustein Bearbeiten Zielsystem Test Ansicht Fenster Hilfe
Adres Deklar Name Typ Anfangswert Aktualwert
1 0.0 in ID WORD W#16#0 W#16#1
2 2.0 in DB_PARAM BLOCK_DB DB 1 DB 2
3 4.0 in RECV_TIME TIME T#0MS T#500MS
4 8.0 in CONN_TIME TIME T#0MS T#5S
5 12.0 in KEEP_ALIVE TIME T#0MS T#0MS
6 16.0 in ENQ_ENR BOOL FALSE TRUE
7 16.1 in DISCONNECT BOOL FALSE FALSE
8 18.0 in REG_KEY STRING [17]
9 38.0 out LICENSED BOOL FALSE FALSE
10 38.1 out BUSY BOOL FALSE FALSE
11 38.2 out CONN_ESTABLISHED BOOL FALSE FALSE
12 38.3 out DONE_NDR BOOL FALSE FALSE
13 38.4 out ERROR BOOL FALSE FALSE
14 40.0 out STATUS_MODBUS WORD W#16#0 W#16#0
15 42.0 out STATUS_CONN WORD W#16#0 W#16#0
16 44.0 out STATUS_FUNC STRING [8]
17 54.0 out IDENT_CODE STRING [18] 'EMHGBBHFMFCJKMBJL2'
18 74.0 in_out UNIT BYTE B#16#0 B#16#0
19 75.0 in_out DATA_TYPE BYTE B#16#0 B#16#0
20 76.0 in_out START_ADDRESS WORD W#16#0 W#16#0
Meldungen
RUN
Please insert the IDENT-CODE here.
The manual contains information how to find out the IDENT-CODE.
Bitte tragen Sie den IDENT-CODE hier ein.
Das Handbuch enthält Informationen, wie Sie den IDENT-CODE ermitteln.
>>> IDENT_CODE <<<
Please insert the License-No. here.
Bitte tragen Sie die Lizenz-Nr. hier ein.
>>> License-No / Lizenz-Nr <<<
S7-OpenModbus/TCP
simatic add on
SOFTWARE

4. Senden Sie das Formular per E-Mail an IT4Industry it4.industry@siemens.com.

   Sie erhalten daraufhin den Freischaltcode für ihre CPU.

Eingabe des Freischaltcodes REG_KEYs

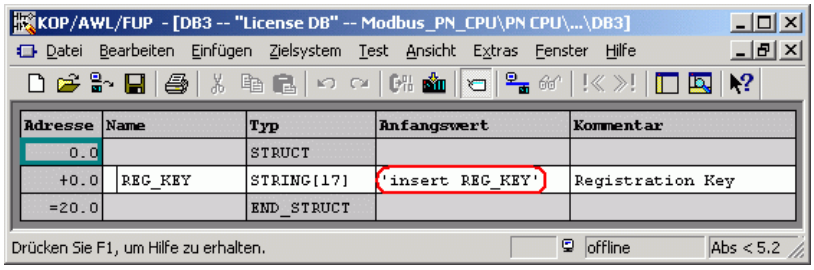
Die Angabe des Freischaltcodes REG_KEY muss an jedem MODBUSPN-Bausteinaufruf erfolgen.

Der REG_KEY sollte in einem Global-DB gespeichert werden, über den alle MODBUSPN-Bausteine den notwendigen Freischaltcode erhalten (siehe nachfolgendes Beispiel).

Für die Eingabe des Freischaltcodes REG_KEY gehen Sie wie folgt vor:

1. Kopieren Sie den vorgefertigten Lizenzierungsbaustein DB3 aus der Bibliothek „Modbus_PN_CPU“ in ihr Projekt. Wird die DB-Nummer bereits im Projekt verwendet, kann der Lizenz-DB auch umbenannt werden.

2. Öffnen Sie den Lizenz-DB und kopieren Sie den von IT4Industry übermittelten 17stelligen Freischaltcode per Copy/Paste in die Spalte ‚Anfangswert’.


KOP/AWL/FUP - [DB3 -- "License DB" -- Modbus_PN_CPU\PN CPU\...\DB3]
Datei Bearbeiten Einfügen Zielsystem Test Ansicht Extras Fenster Hilfe
Adresse Name Typ Anfangswert Kommentar
0.0 STRUCT
+0.0 REG_KEY STRING[17] 'insert REG_KEY' Registration Key
=20.0 END_STRUCT
Drücken Sie F1, um Hilfe zu erhalten.
offline
Abs < 5.2

3. Damit der Freischaltcode nach dem Neuladen der CPU nicht erneut eingegeben werden muss, muss er im Datenbaustein fest eingetragen werden. Wechseln Sie dazu über den Menüpunkt „Ansicht“ -> „Datensicht“ in die Datensicht des DBs. Über den Menübefehl „Bearbeiten“ -> „Datenbaustein initialisieren" werden dann alle Werte der Spalte „Anfangswert" in die Spalte „Aktualwert" übernommen.

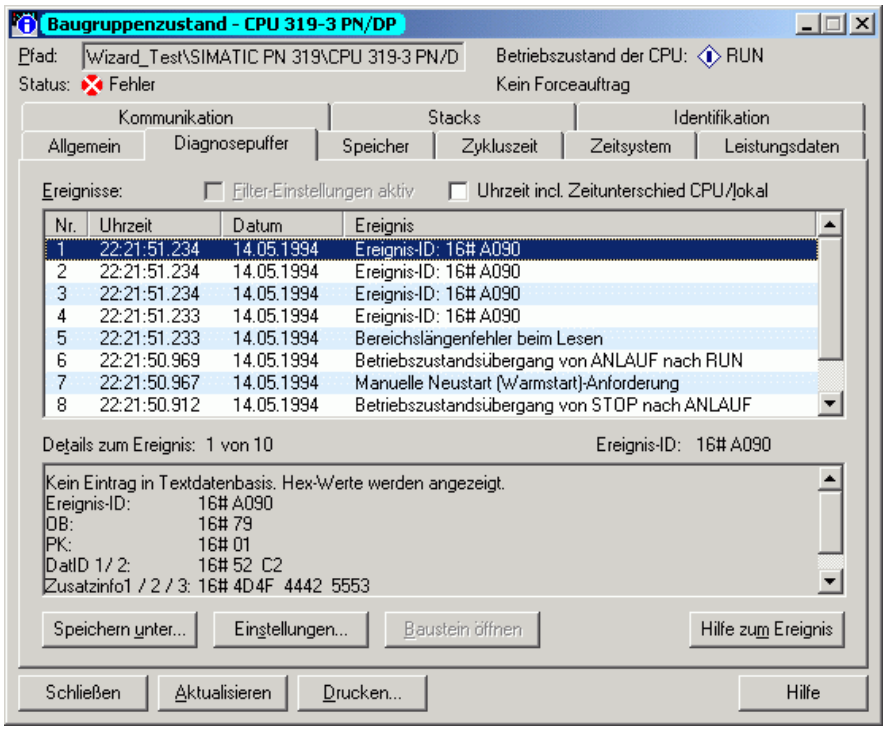
4. Geben Sie im zyklischen OB am Parameter REG_KEY des MODBUSPN-Bausteins den Wert „DB3.REG_KEY“ an.

5. Laden Sie die geänderten Bausteine in die CPU. Die Eingabe des Freischaltcodes kann zur Laufzeit erfolgen, ein Wechsel von STOP -> RUN ist nicht erforderlich.

Der Baustein ist nun für diese CPU lizenziert.

Fehlende oder fehlerhafte Lizenzierung

Ist kein oder ein falscher Freischaltcode eingetragen, blinkt die SF-LED (bei S7-300 und IM151-8) bzw. INTF-LED (bei S7-400) der CPU und es wird zyklisch ein Eintrag in den Diagnosepuffer bezüglich der fehlenden Lizenz vorgenommen. Die Fehlernummer für eine fehlende Lizenz ist W#16#A090.



Die Einträge in den Diagnosepuffer werden über den Programmierfehler-OB OB121 realisiert.

Warnung

Falls der OB121 in der Steuerung fehlt, wird die CPU in den STOP-Zustand gesetzt.

Bei einem fehlenden oder falschen Freischaltcode wird die Modbus-Kommunikation bearbeitet, allerdings wird am Ausgang STATUS_MODBUS stets W#16#A090 „Keine gültige Lizenz vorhanden“ angezeigt.

Wird dieser Fehlercode trotz eingegebenen Freischaltcode angezeigt, dann sollte geprüft werden, ob FC10 “EQ_STRING” in das Projekt eingefügt wurde.

6 Funktionsbaustein MODBUSPN

6.1 Funktionsweise der FB

Allgemeines

Der Funktionsbaustein MODBUSPN erlaubt es eine Kommunikation zwischen einer CPU mit integrierter PN-Schnittstelle und einem Partner, der das OPEN MODBUS/TCP Protokoll unterstützt, aufzubauen.
Es werden die Funktionscodes 1, 2, 3, 4, 5, 6, 15 und 16 unterstützt.

Der FB kann je nach Parametrierung sowohl als Client als auch als Server betrieben werden. In der Betriebsart Server ist der Multiplexbetrieb gemäß MODBUS OPEN/TCP Spezifikation nicht implementiert.

Der Baustein MODBUSPN ruft intern die Bausteine MOD_CLI (FB104), MOD_SERV (105) und TCP_COMM auf. Der Baustein MOD_CLI beinhaltet die Funktionalität des Modbus Clients, der Baustein MOD_SERV realisiert die Funktionalität des Modbus Servers und der Baustein TCP_COMM behandelt das Verbindungsmanagement.

Die Bausteine erfüllen folgende Aufgaben:

- Verbindungs- und Datenhandling unter Verwendung der T-Bausteine aus der Standard-Bibliothek
- MODBUS spezifischen Telegrammheader beim Senden generieren
- Prüfung des MODBUS spezifischen Telegrammheaders beim Empfang
- Prüfung ob die angesprochenen Datenbereiche vorhanden sind
- Exception Telegramme generieren wenn ein Fehler aufgetreten ist (nur bei S7 ist Server)
- Datentransfer von/in den parametrierten DB
- Zeitliche Überwachung des Verbindungsaufbaus- und –abbaus sowie des Empfangs von Daten
- Lizenzprüfung

Online-Hilfe

Für den Funktionsbaustein MODBUSPN steht im SIMATIC-Manager eine Baustein-Online-Hilfe zur Verfügung. Wenn der Baustein markiert und die Taste „F1“ gedrückt wird, wird die Online-Hilfe mit den wichtigsten Informationen zum Baustein geöffnet.

Aufruf des FBs

Der Funktionsbaustein MODBUSPN muss für einen korrekten Programmablauf in 3 OBs eingebaut werden:

- im Anlauf-OB100
- im Programmierfehler-OB121 und
- in einem zyklischen OB (OB1 oder in einem zeitgesteuerten OB, z.B. OB35)

Dabei muss derselbe Instanz-Datenbaustein verwendet werden.
Die anderen in der Bibliothek enthaltenen FBs MOD_CLI, MOD_SERV und TCP_COMM werden unterlagert aufgerufen und dürfen nicht zusätzlich in einem OB aufgerufen werden.

Der gleichzeitige Aufruf des FB MODBUSPN im OB1 und in einem zeitgesteuerten OB (z.B. OB35) ist nicht zulässig.

Der Modbus-Baustein **muss** im OB121 aufgerufen werden. Nähere Informationen dazu erhalten Sie im **Kapitel 5 „Lizenzierung“**.

Anlauf des FBs

Der Funktionsbaustein MODBUSPN wird im OB100 einmal unbedingt aufgerufen.

- Die Initialisierungsparameter müssen entsprechend der Anlagenkonfiguration belegt sein.
- Die Initialisierungsparameter werden in den Instanz-DB übernommen.
- Die Laufzeitparameter werden im Anlauf nicht ausgewertet.
- Die Daten aus MODBUS_PARAM werden auf Plausibilität überprüft.

Zyklischer Betrieb des FBs

Im zyklischen Betrieb wird der FB MODBUSPN im OB1 oder in einem Weckalarm-OB aufgerufen.

- Anhand der Laufzeitparameter werden die Funktionen des Bausteins aktiviert.
- Während ein Auftrag läuft, werden Änderungen an den Laufzeitparametern nicht ausgewertet.
- Initialisierungsparameter werden nicht ausgewertet.

Programmierfehler OB121

Der Baustein MODBUSPN muss im OB121 mit demselben Instanzdatenbaustein wie im Anlauf-OB und im zyklischen OB aufgerufen werden.
Ist der Baustein für diese CPU noch nicht lizenziert, wird OB121 aufgerufen.

Warnung

Falls der OB121 in der Steuerung fehlt, wird die CPU in den STOP-Zustand gesetzt.

Mit dem MODBUSPN-Aufruf im OB121 wird ein Eintrag im Diagnosepuffer der CPU realisiert, der einen Hinweis auf die fehlende Lizenz beinhaltet. Gleichzeitig blinkt die LED SF bzw. INTF der CPU.

Verbindungs-bearbeitung

Den aktiven Verbindungsaufbau kann sowohl der Modbus Client als auch der Modbus Server ausführen. Es wird ausdrücklich empfohlen, dass der Modbus Client den aktiven Part übernimmt. Die Daten hierfür werden aus den Verbindungsparametern im DB MODBUS_PARAM ausgelesen.

Über einen Parameter im Verbindungsparameterblock (*active_est*) wird festgelegt, ob die PN-CPU als aktiver oder als passiver Kommunikationspartner fungieren soll.
Zur Laufzeit wird bei beiden Verbindungstypen, aktiv und passiv, mit der Funktion TCON ein Kommunikationskanal zum Koppelpartner geöffnet.

Der Zeitpunkt des Verbindungsaufbaus wird mit einem Parameter im DB MODBUS_PARAM festgelegt (*connect_at_startup)*.

Der Verbindungsabbau wird mit dem Parameter DISCONNECT am FB MODBUSPN geregelt.

Mehrere Koppelpartner

Eine PN-CPU kann Verbindungen zu mehreren Koppelpartnern aufbauen. Damit die Verbindungsbearbeitung und der Datentransfer problemlos abgearbeitet werden können, ist eine strikte Trennung der jeweiligen Verbindungen zu den verschiedenen Koppelpartnern notwendig.

Das heißt **pro Verbindung** wird benötigt:

- Ein Verbindungsparameterblock und die zugehörigen Modbusparameter im DB MODBUS_PARAM
- Aufruf von FB MODBUSPN in OB100
- Aufruf von FB MODBUSPN in OB121
- Aufruf von FB MODBUSPN in OB1 oder einem Weckalarm-OB

Dabei erhalten die Aufrufe des Funktionsbausteins MODBUSPN im OB100, OB121 und im OB1 für die jeweilige Verbindung denselben Instanz-Datenbaustein. Für jede weitere Verbindung wird ein anderer Instanz-Datenbaustein verwendet.

Auftragsanstoß S7 ist Client

Durch einen positiven Flankenwechsel am Triggereingang ENQ_ENR wird ein Auftrag initiiert. Abhängig von den Eingangsparametern UNIT, DATA_TYPE, START_ADDRESS, LENGTH, TI und WRITE_READ wird ein MODBUS Anforderungstelegramm generiert und zur Partnerstation über die TCP/IP Verbindung gesendet. Der Client wartet die parametrierte Zeit RECV_TIME auf eine Antwort vom Server.
Kommt es zu einer Zeitüberschreitung (keine Antwort vom Server) wird der aktivierte Auftrag mit Fehler beendet. Ein neuer Auftrag kann initiiert werden.

Nach dem Empfang des Antworttelegramms wird eine Plausibilitätsprüfung durchgeführt. Verläuft diese positiv, werden die erforderlichen Aktionen durchgeführt und der Auftrag wird ohne Fehler beendet, der Ausgang DONE_NDR wird gesetzt. Wurden bei der Prüfung Fehler erkannt, wird der Auftrag mit Fehler beendet, das Bit ERROR wird gesetzt und eine Fehlernummer in STATUS_MODBUS angezeigt.

Aktivierung des FBs S7 ist Server

Durch einen positiven Pegel am Triggereingang ENQ_ENR ist der FB zum Empfang eines Anforderungstelegramms vom Client bereit. Der Server verhält sich dabei passiv und wartet auf ein Telegramm vom Client. Das empfangene Telegramm wird überprüft. Verläuft die Prüfung positiv, wird das Anforderungstelegramm beantwortet. Der beendete Telegrammverkehr wird dem Anwender mitgeteilt indem das Bit DONE_NDR gesetzt wird. Zu diesem Zeitpunkt wird die ausgeführte Funktion an den Ausgängen UNIT, DATA_TYPE, START_ADDRESS, LENGTH, TI und WRITE_READ angezeigt.

Ein fehlerhaftes Anforderungstelegramm bewirkt eine Fehlermeldung. Das Bit ERROR wird gesetzt, im STATUS_MODBUS wird die Fehlernummer angezeigt und die Anforderung des Client wird nicht beantwortet.

6.2 Parameter des Funktionsbausteins MODBUSPN

Parameter	Dekl.	Typ	Beschreibung	Wertebereich	Init
ID	IN	WORD	Verbindungs-ID, muss identisch sein mit dem zugehörigen Parameter *id* im Parameter-DB MODBUS_PARAM	1 bis 4095 W#16#1 bis W#16#FFF	ja
DB_PARAM	IN	BLOCK_DB	Nummer des Parameter-DBs MODBUS_PARAM	CPU abhängig	ja
RECV_TIME	IN	TIME	Überwachungszeit für den Empfang von Daten vom Koppelpartner Die minimal einstellbare Zeit ist 20 ms	T#20ms bis T#+24d20h31m23s647ms	nein
CONN_TIME	IN	TIME	Überwachungszeit für den Verbindungsaufbau bzw. –abbau Die minimal einstellbare Zeit ist 100 ms	T#100ms bis T#+24d20h31m23s647ms	nein
KEEP_ALIVE	IN	TIME	nicht verwendet		
ENQ_ENR	IN	BOOL	S7 ist Client: Auftragsanstoß bei positiver Flanke S7 ist Server: Empfangsbereit bei positivem Pegel	TRUE FALSE	nein
DISCONNECT	IN	BOOL	S7 ist Client: TRUE: nach Empfang des Antworttelegramms wird die Verbindung abgebaut S7 ist Server: TRUE: bei ENQ_ENR = FALSE soll Verbindung abgebaut werden	TRUE FALSE	nein
REG_KEY	IN	STRING [17]	Registrierungsschlüssel (Freischaltcode) für die Lizenzierung	Character	nein
LICENSED	OUT	BOOL	Lizenzzustand des Bausteins Baustein ist lizenziert Baustein ist nicht lizenziert	 TRUE FALSE	nein
BUSY	OUT	BOOL	Bearbeitungszustand der T-Funktionen (TCON, TDISCON, TSEND oder TRCV) in Bearbeitung nicht in Bearbeitung	 TRUE FALSE	nein
CONN_ESTABLISHED	OUT	BOOL	Verbindung ist aufgebaut Verbindung ist abgebaut	TRUE FALSE	nein
DONE_NDR	OUT	BOOL	S7 ist Client: TRUE: aktivierter Auftrag wurde fehlerfrei beendet S7 ist Server: TRUE: Anforderung vom Client wurde ausgeführt und beantwortet	TRUE FALSE	nein

Parameter	Dekl.	Typ	Beschreibung	Wertebereich	Init
ERROR	OUT	BOOL	Es ist ein Fehler aufgetreten. Es ist kein Fehler aufgetreten.	TRUE FALSE	nein
STATUS_ MODBUS	OUT	WORD	Fehlernummer für Protokollfehler bei der Bearbeitung der Modbustelegramme	0 bis FFFF	nein
STATUS_ CONN	OUT	WORD	Fehlernummer für Verbindungsfehler bei der Bearbeitung der T-Bausteine (TCON, TSEND, TRCV, TDISON)	0 bis FFFF	nein
STATUS_FUNC	OUT	STRING [8]	Name der Funktion, die den Fehler an STATUS_MODBUS bzw. STATUS_CONN verursacht hat	Character	nein
IDENT_CODE	OUT	STRING [18]	Identifikationsnummer für die Lizenzierung Mit dieser Kennung können Sie den Freischaltcode REG_KEY für Ihre Lizenz beantragen.	Character	nein
UNIT	IN/ OUT	BYTE	Unit Identifier (INPUT bei CLIENT Funktion, OUTPUT bei SERVER Funktion)	0 bis 255 B#16#0 bis B#16#FF	nein
DATA_TYPE	IN/ OUT	BYTE	zu bearbeitender Datentyp: (INPUT bei CLIENT Funktion, OUTPUT bei SERVER Funktion) Coils Inputs Holding Register Input Register	 1 2 3 4	nein
START_ ADDRESS	IN/ OUT	WORD	MODBUS Startadresse (INPUT bei CLIENT Funktion, OUTPUT bei SERVER Funktion)	0 bis 65535 W#16#0000 bis W#16#FFFF	nein
LENGTH	IN/ OUT	WORD	Anzahl der zu bearbeitenden Werte (INPUT bei CLIENT Funktion, OUTPUT bei SERVER Funktion) <u>Coils</u> Lesende Funktion Schreibende Funktion <u>Inputs</u> Lesende Funktion <u>Holding Register</u> Lesende Funktion Schreibende Funktion <u>Input Register</u> Lesende Funktion	 1 bis 2000 1 bis 800 1 bis 2000 1 bis 125 1 bis 100 1 bis 125	nein
TI	IN/ OUT	WORD	Transaction Identifier (INPUT bei Client Funktion, OUTPUT bei Server Funktion)	0 bis 65535 W#16#0 to W#16#FFFF	Nein
WRITE_ READ	IN/ OUT	BOOL	Schreibzugriff oder Lesezugriff (INPUT bei CLIENT Funktion, OUTPUT bei SERVER Funktion)	TRUE FALSE	Nein

Allgemeines

Die Parameter des FB MODBUSPN gliedern sich in zwei Gruppen:

- Initialisierungsparameter
- Laufzeitparameter

Die **Initialisierungsparameter** werden nur beim Aufruf im OB100 ausgewertet und in den Instanz-DB übernommen. Die Initialisierungsparameter sind in der obigen Tabelle in der Spalte „INIT“ mit „ja“ gekennzeichnet.
Eine Änderung der Initialisierungsparameter während des laufenden Betriebs hat keine Auswirkung. Nach einer Änderung dieser Parameter z.B. im Testbetrieb muss der Instanz-DB (I-DB) durch STOP -> RUN der CPU neu initialisiert werden.

Laufzeitparameter können im zyklischen Betrieb verändert werden.
In der Betriebsart „S7 ist Client“ ist es jedoch nicht sinnvoll die Eingangsparameter zu ändern während ein Auftrag läuft. Mit den Vorbereitungen für den nächsten Auftrag und den damit verbundenen Änderungen der Parameter sollte gewartet werden, bis der vorherige Auftrag mit DONE_NDR oder ERROR beendet wurde.

In der Betriebsart „S7 ist Server“ dürfen die Ausgangsparameter nur bei gesetztem DONE_NDR ausgewertet werden.

Die Ausgangsparameter sind **dynamische Anzeigen** und stehen somit nur **1 CPU-Zyklus** an. Sie müssen für eine eventuelle Weiterverarbeitung oder eine Anzeige in der Variablentabelle in andere Speicherbereiche kopiert werden.

Wertebereiche

Bei den Wertebereichen für die verschiedenen Parameter sind ggf. auch CPU-spezifische Einschränkungen zu beachten.

ID

Für jede Verbindung von der PN-CPU zu einem Kommunikationspartner wird eine Verbindungs-ID benötigt. Bei mehreren Kommunikationspartnern ist für jede logische Verbindung eine andere Verbindungs-ID zu verwenden.
Diese Verbindungs-ID wird im Verbindungsparameterblock projektiert, welcher im Parameterdatenbaustein MODBUS_PARAM enthalten ist. Die Verbindungs-ID beschreibt eindeutig die Verbindung von der CPU zum Koppelpartner und kann Werte von 1 bis 4095 annehmen.
Die Verbindungs-ID aus dem Verbindungsparameterblock ist hier einzutragen und muss CPU-weit eindeutig sein.

DB_PARAM

Der Parameter DB_PARAM bezeichnet die Nummer des Datenbausteins MODBUS_PARAM. In diesem Parameterdatenbaustein sind die verbindungs- und modbusspezifischen Parameter hinterlegt, die für die Kommunikation zwischen der PN-CPU und dem Koppelpartner notwendig sind.

Der Wertebereich für diesen Parameter ist CPU-abhängig. Die DB-Nummer 0 ist nicht zulässig, da diese für das System reserviert ist.
Die Eingabe der DB-Nummer erfolgt im Klartext in der Form „DBxy“.

Sollen mehrere Verbindungen realisiert werden, kann der Parameterdaten-baustein die dazu notwendigen Parameter aller Verbindungen in sequentieller Abfolge enthalten. Es ist ebenso möglich für jede Verbindung einen separaten Parameterdatenbaustein zu erstellen.

RECV_TIME

Die Überwachungszeit RECV_TIME überwacht den Empfang der Daten vom Koppelpartner. Der Minimalwert beträgt 20 ms.

Wenn in der Betriebsart „**S7 ist Client**“ die RECV_TIME auf < 20 ms gesetzt wird, erscheint eine entsprechende Fehlermeldung und der aktivierte Auftrag wird abgewiesen. Bei Überschreitung der Überwachungszeit wird der aktivierte Auftrag mit Fehler beendet.
Wenn in der Betriebsart „**S7 ist Server**“ die RECV_TIME auf < 20 ms gesetzt wird, wird der Defaultwert von 1,2 s verwendet. Bei Überschreitung der Überwachungszeit wird ein Fehler gemeldet. Die RECV_TIME überwacht die Laufzeit des TCP-Streams. Die Pause zwischen einzelnen Request vom Client wird dabei nicht berücksichtigt.

CONN_TIME

Die CONN_TIME gibt die Zeit für die Überwachung des Verbindungsaufbaus bzw. -abbaus an. Der minimale Wert ist 100 ms.

Konnte innerhalb der parametrierten Überwachungszeit die Verbindung nicht erfolgreich auf- bzw. abgebaut werden, erscheint eine entsprechende Fehlermeldung am Ausgang STATUS_CONN.

Wenn in der Betriebsart „**S7 ist Server**“ die CONN_TIME auf < 100 ms gesetzt wurde, wird der Defaultwert von 5 s verwendet.

In der Betriebsart „**S7 ist Client**“ wird bei *connect_at_startup* = TRUE eine zu gering parametrierte CONN_TIME ebenfalls auf 5 s gesetzt. Im zyklischen Betrieb wird bei einer zu kleinen CONN_TIME eine Fehlermeldung ausgegeben und der aktivierte Auftrag abgewiesen.

ENQ_ENR

Betriebsart „**S7 ist Client**“:
Mit einer positiven Flanke wird der Datentransfer initiiert. Mit den Werten der Eingangsparameter UNIT, DATA_TYPE, START_ADDRESS, LENGTH, TI und WRITE_READ wird das Anforderungstelegramm generiert. Ein neuer Auftrag kann nur gesendet werden, wenn der vorherige mit DONE_NDR oder ERROR abgeschlossen wurde.
Wenn die Verbindung nicht aufgebaut ist (CONN_ESTABLISHED = FALSE), wird zuerst die Verbindung aufgebaut und dann der Datentransfer ausgeführt.

Betriebsart „**S7 ist Server**“:
Mit einem positiven Pegel am Eingang wird der FB aktiviert. Es können Telegramme vom Client empfangen werden.
Falls die Verbindung bei gesetztem ENQ_ENR nicht aufgebaut ist (CONN_ESTABLISHED = FALSE), wird der Verbindungsaufbau aktiviert.
Wechselt während des laufenden Betriebs ENQ_ENR von TRUE auf FALSE wird abhängig von der Einstellung am Parameter DISCONNECT ggf. die Verbindung abgebaut.
Bei einem nicht gesetzten Eingang ENQ_ENR und einer bestehenden Verbindung werden die empfangenen Daten verworfen.

DISCONNECT

In der Betriebsart „**S7 ist Client**“ wird mit DISCONNECT = TRUE festgelegt, dass die Verbindung nach dem Datentransfer abgebaut werden soll.

In der Betriebsart „**S7 ist Server**“ wird mit DISCONNECT = TRUE die Verbindung abgebaut, wenn der Parameter ENQ_ENR auf FALSE gesetzt wird.

Dieser Parameter ist ein Laufzeitparameter und kann entsprechend der Anforderung beliebig gesetzt bzw. zurückgesetzt werden.

FREE_CODE

Mit der korrekten Eingabe des Freischaltcode an diesen Parameter wird der Baustein lizenziert und die Modbus-Kommunikation kann ohne Einschränkungen genutzt werden.

Weitere Informationen entnehmen Sie bitte dem **Kapitel 5 „Lizenzierung“**.

BUSY

Wenn dieser Ausgang gesetzt ist, ist eine der T-Funktionen TCON, TDISCON, TSEND oder TRCV in Bearbeitung.

CONN_ ESTABLISHED

Mit CONN_ESTABLISHED = TRUE wird angezeigt, dass eine Verbindung zum Koppelpartner besteht und Daten übertragen werden können.

Ist CONN_ESTABLISHED = FALSE besteht keine Verbindung zum Koppelpartner.

DONE_NDR

In der Betriebsart „**S7 ist Client**“ wurde der aktivierte Auftrag fehlerfrei beendet. Bei einer lesenden Funktion wurden die Antwortdaten vom Server bereits im DB eingetragen, bei einer schreibenden Funktion wurde vom Server die Antwort auf das Anforderungstelegramm erhalten.

In der Betriebsart „**S7 ist Server**“ zeigt der Ausgang einen fehlerfrei beendeten Telegrammverkehr mit dem Client an. In den Parametern UNIT, DATA_TYPE, START_ADDRESS, LENGTH, TI und WRITE_READ werden die Auftragsparameter des Client angezeigt. Diese Ausgänge sind nur gültig so lange DONE_NDR gesetzt ist.

ERROR

Wenn dieser Ausgang gesetzt ist, wurde ein Fehler erkannt.

In der Betriebsart „**S7 ist Client**“ wurde der aktivierte Auftrag mit Fehler abgeschlossen. Die zugehörige Fehlernummer wird in den Ausgängen STATUS_MODBUS oder STATUS_CONN angezeigt.

In der Betriebsart „**S7 ist Server**“ wurde bei einem Anforderungstelegramm des Client oder beim Senden des Antworttelegramms ein Fehler erkannt. Die zugehörige Fehlernummer wird in den Ausgängen STATUS_MODBUS oder STATUS_CONN angezeigt.

STATUS_MODBUS

Der Ausgang STATUS_MODBUS zeigt bei gesetztem ERROR die Fehlernummer bezüglich der Modbus-Telegrammbearbeitung an. Die Fehlernummern sind im Kapitel 7 beschrieben.

STATUS_CONN

Der Ausgang STATUS_CONN zeigt bei gesetztem ERROR die Fehlernummer bezüglich der Verbindungsbearbeitung an. Die Fehlernummern sind im Kapitel 7 bzw. in den Bausteinonlinehilfen der T-Bausteine TCON, TDISCON, TSEND und TRCV beschrieben.

Zusätzlich werden in STATUS_CONN auch Meldungen wie „Auftrag in Bearbeitung“ angezeigt. In diesem Fall ist das ERROR-Bit nicht gesetzt.

STATUS_FUNC

An diesem Parameter wird als Character-String der Name der Funktion angezeigt, die den aufgetretenen Fehler verursacht hat.

LICENSE_ID

Nach dem Anlauf der CPU wird an diesem Parameter eine 18stellige Identifikationskennung angezeigt, mit dem der Lizenzierungscode REG_KEY (= Freischaltungscode) für die Modbus-Kommunikation beantragt wird.

Weitere Informationen entnehmen Sie bitte dem **Kapitel 5 „Lizenzierung“**.

UNIT

In der Funktion „**S7 ist Client**“ ist der Parameter UNIT ein Eingangsparameter. Dieser Eingang ist den Anforderungen entsprechend zu setzen. Der FB übernimmt diesen Wert in das Anforderungstelegramm und überprüft ihn beim Empfang der Antwort.

In der Funktion „**S7 ist Server**“ ist der Parameter UNIT ein Ausgangsparameter. Der FB übernimmt den Wert aus dem Anforderungstelegramm in die Antwort. Der Ausgang wird bei beendetem Auftrag mit dem empfangenen Wert gesetzt.

DATA_TYPE

Der Parameter DATA_TYPE zeigt an, welcher Modbus-Datentyp mit dem aktuellen Telegramm bearbeitet wird. Es sind folgende Werte zulässig:

Coils	B#16#1
Inputs	B#16#2
Holding Register	B#16#3
Input Register	B#16#4

In der Betriebsart „**S7 ist Client**“ handelt es sich um einen Eingangs-, in der Betriebsart „**S7 ist Server**“ um einen Ausgangsparameter.

Die unterschiedlichen Datentypen haben einen direkten Zusammenhang mit den verwendeten Funktionscodes.

Datentyp	DATA_ TYPE	Funktion	Länge	single_ write	Funk-tions-code
Coils	1	lesen	beliebig	irrelevant	1
Coils	1	schreiben	1	TRUE	5
Coils	1	schreiben	1	FALSE	15
Coils	1	schreiben	>1	irrelevant	15
Inputs	2	lesen	beliebig	irrelevant	2
Holding Register	3	lesen	beliebig	irrelevant	3
Holding Register	3	schreiben	1	TRUE	6
Holding Register	3	schreiben	1	FALSE	16
Holding Register	3	schreiben	>1	irrelevant	16
Input Register	4	lesen	beliebig	irrelevant	4

START_ADDRESS

Der Parameter START_ADDRESS bestimmt die erste MODBUS-Adresse, die geschrieben bzw. gelesen wird.

In der Betriebsart „**S7 ist Client**“ handelt es sich um einen Eingangs-, in der Betriebsart „**S7 ist Server**“ um einen Ausgangsparameter.

LENGTH

Der Parameter LENGTH bestimmt die Anzahl der MODBUS-Werte, die geschrieben bzw. gelesen wird.

In der Betriebsart „**S7 ist Client**“ handelt es sich um einen Eingangs-, in der Betriebsart „**S7 ist Server**“ um einen Ausgangsparameter.

Bei lesenden Funktionen sind pro Telegramm für Holding und Input Registern maximal 125 Register möglich. Für Coils und Inputs sind maximal 2000 Bits möglich. Bei schreibenden Funktionen beträgt bei Holding Register die maximale Anzahl 100 Register und bei Coils 800 Bits.
Die mit einem Anforderungstelegramm bearbeiteten Register bzw. Bitwerte müssen innerhalb eines DBs liegen.

TI

Der Parameter TI, Transaction Identifier, wird gemäß MODBUS Spezifikation vom Server aus dem Anforderungstelegramm in das Antworttelegramm umkopiert.

In der Betriebsart „**S7 ist Client**“ handelt es sich um einen Eingangsparameter. Der FB übernimmt diesen Wert in das Anforderungstelegramm und überprüft ihn beim Empfang der Antwort.
In der Betriebsart „**S7 ist Server**“ handelt es sich um einen Ausgangsparameter. Der FB übernimmt den Wert aus dem Anforderungstelegramm in die Antwort.

Der Transaction Identifier dient zur Erkennung von Telegrammen bzw. der eindeutigen Zuordnung von den Anfragen zu den Antworten. Der FB MODBUS kann diese Zuordnung nur vornehmen, wenn der TI auch bei jedem Telegramm verändert wird. Nur dann ist eine zuverlässige Funktion des FB gewährleistet.

Deshalb wird empfohlen, den TI bei jedem Request um 1 zu erhöhen.

WRITE_READ

Dieser Parameter definiert ob eine lesende oder schreibende Funktion ausgeführt werden soll. Hat der Ein-/Ausgang den Wert FALSE, handelt es sich um eine lesende Funktion. Der Wert TRUE definiert eine schreibende Funktion.

Es können nur Holding Register und Coils beschrieben werden. Input Register und Inputs lassen sich lediglich lesen.

In der Betriebsart **„S7 ist Client“** handelt es sich um einen Eingangs-, in der Betriebsart **„S7 ist Server“** um einen Ausgangsparameter.

6.3 Adressabbildung

Interpretation der Modbus Adressen

Das MODBUS Datenmodell basiert auf einer Reihe von Speicherbereichen die unterschiedliche Charakteristiken haben. Die Unterscheidung dieser Speicherbereiche erfolgt bei einigen Systemen, z.B. MODICON PLCs über die Register- bzw. Bitadresse. So wird z.B. das Holding Register mit Offset 0 als Register 40001 bezeichnet (Speichertyp 4xxxx, Reference 0001).

Es führt immer wieder zur Verwirrung, weil in manchen Handbüchern die Registeradresse des Application Layers und in anderen die tatsächlich im Protokoll übertragene Register-/Bitadresse beschrieben und gemeint ist.

Der FB MODBUS verwendet bei seinen Parametern START_x und START_ADDRESS die **tatsächlich übertragene Modbusadresse**. Es können also mit jedem Funktionscode Register-/Bitadressen von 0000_H bis $FFFF_H$ übertragen werden.

Beispiel: Parametrierung der Datenbereiche

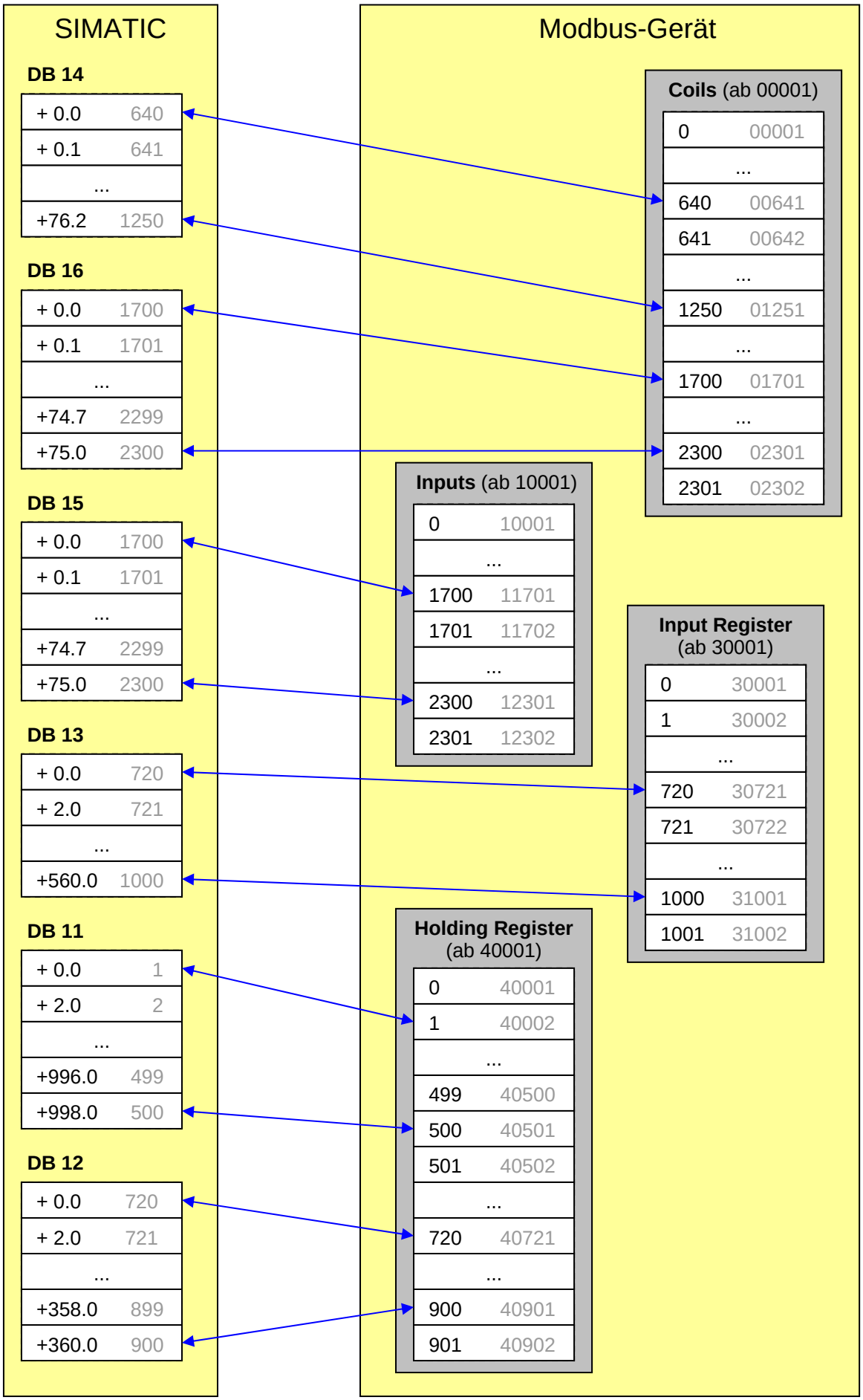
data_type_1	B#16#3	Holding Register
db_1	W#16#B	DB 11
start_1	W#16#1	Anfangsadresse: 1
end_1	W#16#1F4	Endadresse: 500
data_type_2	B#16#3	Holding Register
db _2	W#16#C	DB 12
start _2	W#16#2D0	Anfangsadresse: 720
end _2	W#16#384	Endadresse: 900
data_type_3	B#16#4	Input Register
db _3	W#16#D	DB 13
start _3	W#16#2D0	Anfangsadresse: 720
end _3	W#16#3E8	Endadresse: 1000
data_type_4	B#16#0	Nicht verwendet
db _4	0	0
start _4	0	0
end _4	0	0
data_type_5	B#16#1	Coils
db _5	W#16#E	DB 14
start _5	W#16#280	Anfangsadresse: 640
end _5	W#16#4E2	Endadresse: 1250
data_type_6	B#16#2	Inputs
db _6	W#16#F	DB 15
start _6	W#16#6A4	Anfangsadresse:1700
end _6	W#16#8FC	Endadresse: 2300
data_type_7	B#16#1	Coils
db _7	W#16#10	DB 16
start _7	W#16#6A4	Anfangsadresse: 1700
end _7	W#16#8FC	Endadresse: 2300
data_type_8	B#16#0	Nicht verwendet
db _8	0	0
start _8	0	0
end _8	0	0

Adressabbildung

Im folgenden Bild sehen Sie die Gegenüberstellung der SIMATIC-Speicherbereiche mit der registerorientierten Speicheraufteilung der Modbus-Geräte. Dabei wird auf obige Parametrierung Bezug genommen.

Die Modbus-Adressen die schwarz dargestellt sind, betreffen den Data Link Layer, die grau dargestellten den Application Layer.

Die schwarz dargestellten SIMATIC-Adressen sind der Offset im DB. Zusätzlich sind in grau die Modbus Registernummern eingetragen.


SIMATIC
Modbus-Gerät
DB 14
+ 0.0 640
+ 0.1 641
...
+76.2 1250
DB 16
+ 0.0 1700
+ 0.1 1701
...
+74.7 2299
+75.0 2300
DB 15
+ 0.0 1700
+ 0.1 1701
...
+74.7 2299
+75.0 2300
DB 13
+ 0.0 720
+ 2.0 721
...
+560.0 1000
DB 11
+ 0.0 1
+ 2.0 2
...
+996.0 499
+998.0 500
DB 12
+ 0.0 720
+ 2.0 721
...
+358.0 899
+360.0 900
Coils (ab 00001)
0 00001
...
640 00641
641 00642
...
1250 01251
...
1700 01701
...
2300 02301
2301 02302
Inputs (ab 10001)
0 10001
...
1700 11701
1701 11702
...
2300 12301
2301 12302
Input Register (ab 30001)
0 30001
1 30002
...
720 30721
721 30722
...
1000 31001
1001 31002
Holding Register (ab 40001)
0 40001
1 40002
...
499 40500
500 40501
501 40502
...
720 40721
...
900 40901
901 40902

6.4 Vom FB verwendete Daten und Standardfunktionen

Instanz-DB

Der Funktionsbaustein MODBUSPN speichert seine Daten in einem Instanz-DB. Dieser Instanz-DB wird beim ersten Aufruf des FB durch STEP7 generiert.

Der Instanz-Datenbaustein enthält Parameter vom Typ Input, Output, Input/Output sowie statische Variablen, die er für seinen Ablauf benötigt. Diese Variablen sind remanent und behalten zwischen den FB-Aufrufen ihre Gültigkeit. Über die Variablen wird der interne Ablauf des FBs gesteuert.

Speicherbedarf des Instanz-DBs:

Instanz-DB	Arbeitsspeicher	Ladespeicher
MODBUSPN	830 Byte	1852 Byte

Lokale Variablen

Für den FB werden 96 Byte Lokaldaten benötigt. Hinzu kommen die Lokaldaten der unterlagerten FBs MOD_CLI (110 Byte) oder MOD_SERV (84 Byte), des TCP_COMM 2 Byte und von einem TCON, TDISCON, TSEND oder TRCV, je nach Baustein bis zu 28 Byte. Das ergibt maximal 236 Byte Lokaldaten für einen FB MODBUSPN-Aufruf.

Parameter-DB

Die verbindungs- und modbusspezifischen Parameter werden in dem Parameter-DB MODBUS_PARAM gespeichert.

Zeiten

Der Funktionsbaustein verwendet keine Timer.

Merker

Der Funktionsbaustein verwendet keine Merker.

Standard-FBs zur Verbindungs-bearbeitung

Der im FB MODBUSPN aufgerufene FB TCP_COMM verwendet die Bausteine TCON und TDISCON aus der Standard-Bibliothek für den Auf- und Abbau der Verbindungen zwischen der CPU und dem Kommunikationspartner.

Standard-FBs zur Datenübertragung

Der im FB MODBUSPN aufgerufene FB TCP_COMM verwendet die Bausteine TSEND und TRCV aus der Standard-Bibliothek für die Datenübertragung zwischen der CPU und dem Kommunikationspartner.

MODBUSPN: SFCs für sonstige Funktionen

Der FB MODBUSPN verwendet folgende SFCs aus der Standardbibliothek:

- FC10 “EQ_STRING”
- SFC6 „RD_SINFO“
- SFC20 „BLKMOV“
- SFC24 “TEST_DB”
- SFC51 “RDSYST”
- SFC52 “WR_USMSG”

MOD_CLI and MOD_SERV: SFCs für sonstige Funktionen

Die FBs MOD_CLI und MOD_SERV verwenden folgende SFCs aus der Standardbibliothek:

- SFC20 „BLKMOV“
- SFC24 “TEST_DB”

TCP_COMM: SFCs für sonstige Funktionen

Der FB TCP_COMM verwendet außer den T-Bausteinen den folgenden SFB aus der Standardbibliothek:

- SFB4 „TON“

6.5 Umbenennen von Standardfunktionen

Veranlassung

Falls in Ihrem Projekt die Nummern der Standardfunktionen bereits verwendet werden oder ist der Nummernbereich für andere Applikationen reserviert, können Sie die intern aufgerufenen Funktionsbausteine FB63, FB64, FB65 und FB66 des FBs TCP_COMM oder die aufgerufenen Bausteine MOD_CLI, MOD_SERV und TCP_COMM umbenennen.

Die Systemfunktionen SFC6, SFC20, SFC24, SFC51 und SFC52 sowie der Systemfunktionsbaustein SFB4 können nicht umbenannt werden.

Umbenennen

Gehen Sie zum Umbenennen für die FBs folgendermaßen vor:

1. Ändern Sie in Ihrem Programm im SIMATIC Manager die Nummern der FBs in die gewünschten Nummern.
2. Ziehen Sie diese Änderungen in der Symboltabelle nach.

Ein Ändern der FB-Nummern über Extras -> Umverdrahten ist nicht notwendig.

7 Diagnose

Diagnose-funktionen

Die Diagnosefunktionen der PN-CPU erlauben Ihnen ein schnelles Lokalisieren aufgetretener Fehler. Folgende Diagnosemöglichkeiten stehen zur Verfügung:

- Diagnose über die Anzeigeelemente der CPU
- Diagnose über die Ausgänge STATUS_MODBUS und STATUS_CONN des MODBUSPN-Funktionsbausteins.

Anzeigeelemente (LED)

Die Anzeigeelemente informieren Sie über den Betriebszustand bzw. über mögliche Fehlerzustände der CPU. Die Anzeigeelemente geben Ihnen einen ersten Überblick über aufgetretene interne bzw. externe Fehler sowie schnittstellenspezifische Fehler.

STATUS-Ausgänge des FB MODBUSPN

Für eine Fehlerdiagnose besitzt der Funktionsbaustein MODBUSPN 3 STATUS-Ausgänge.

Der Ausgang STATUS_MODBUS gibt Fehlermeldungen bezüglich der modbusspezifischen Telegrammbearbeitung aus.

Der Ausgang STATUS_CONN gibt Zustandsanzeigen und Fehlermeldungen bezüglich der Verbindungsbearbeitung aus.

Der Ausgang STATUS_FUNC gibt den Namen der Funktion aus, die den Fehler verursacht hat.

Durch Lesen der STATUS-Ausgänge erhalten Sie allgemeine Aussagen zu Fehlern, die bei der Kommunikation aufgetreten sind. Die STATUS-Parameter können Sie im Anwenderprogramm auswerten.

7.1 Diagnose über die Anzeigeelemente der CPU

Anzeigefunktionen

Über die Anzeigeelemente der CPU erhalten Sie Auskunft über den Baugruppenzustand. Zu unterscheiden sind folgende Anzeigefunktionen:

- **Sammelstörungsanzeigen**

  PN-CPU 300
  - SF Sammelfehler
  Blinkt diese LED, dann ist der Modbus-Baustein noch nicht lizenziert.
  Weitere Informationen entnehmen Sie dem **Kapitel 5 „Lizenzierung“**.

  PN-CPU 400
  - INTF Sammelfehler
  Blinkt diese LED, dann ist der Modbus-Baustein noch nicht lizenziert.
  Weitere Informationen entnehmen Sie dem **Kapitel 5 „Lizenzierung“**.

- **Sonderanzeigen**

  PN-CPU 300, PN-CPU 400 und IM 151-8 PN/DP CPU:
  - RX/TX ein Telegramm wird über die Schnittstelle übertragen

Eine detaillierte Beschreibung der Anzeigeelemente finden Sie im jeweiligen Gerätehandbuch der CPUs.

7.2 Überprüfungen durch den FB MODBUSPN

Im Anlauf

- Eindeutigkeit der Parameter *id* in den Verbindungsparametern des DB MODBUS_PARAM
- Parametergruppe *data_type_x*, *db_x*, *start_x*, *end_x*. (x = 1 bis 8)

    1. Mit *data_type_x* = 0 wird der Modbusbereich abgeschaltet und nicht weiter überprüft. Mindestens der erste Bereich (*_1*) muss parametriert werden.
    2. Prüfung von *data_type_x* auf Gültigkeit
    3. Prüfung von *db_x* <> 0
    4. Test *end_x* >= *start_x*
    5. Modbusadressen für den gleichen Datentyp, die in zwei *db_x* definiert werden, führen zu einer Fehlermeldung (Überlappung).

Ein Fehler im Anlauf bewirkt, dass der Ausgang ERROR gesetzt bleibt und im zyklischen Betrieb keine Modbus-Kommunikations-Aufträge ausgeführt werden können. Es ist eine Korrektur der Parametrierung und STOP → RUN an der CPU erforderlich.

Zyklischer Betrieb S7 ist Client

Überprüfungen bei Aufruf des FB:

- Gültige Lizenz
- DB MODBUS_PARAM hat die benötigte Länge
- Wertebereich der Überwachungszeiten RECV_TIME und CONN_TIME
- Wertebereich von START_ADDRESS
- Wertebereich von LENGTH
- Bei der Bearbeitung eines Auftrages wird überprüft, ob der durch die Modbusadresse angesprochene Datenbaustein in der erforderlichen Länge vorhanden ist. Die DB-Nummer darf nicht 0 und nicht identisch mit MODBUS_PARAM oder dem zugeordneten Instanz-DB sein.
- Eingang des Antworttelegramms innerhalb der Überwachungszeit

  Die Überwachungszeit kann auch ablaufen, wenn weniger Daten als im MODBUS-Telegrammheader angegeben empfangen werden. Es kann dann zu Folgefehlern mit Telegrammverlust kommen.

Überprüfungen im Antworttelegramm:

- Empfangener Transaction Identifier ist gleich dem gesendeten
- Protocol Identifier = 0
- Länge zwischen 3 und 253
  Außerdem wird die Länge aus dem Header des Antworttelegramms einer Plausibilitätsprüfung bezüglich des Auftrages unterzogen.
- Empfangene UNIT ist gleich gesendeter
- Empfangener Funktionscode ist gleich dem gesendeten
- Antwort ist ein Exception Code Telegramm
- Bei Schreibaufträgen wird überprüft, dass die Anfangsadresse und die Register-/Bitanzahl mit dem Anforderungstelegramm übereinstimmen.
- Im Antworttelegramm bei FC 5 und FC 6 ist das Echo der Anforderung korrekt.

Zyklischer Betrieb S7 ist Server

- Gültige Lizenz
- Eingang des zweiten Teils des Anforderungstelegramms ist innerhalb der Überwachungszeit

  Die Überwachungszeit kann auch ablaufen, wenn weniger Daten als im MODBUS-Telegrammheader angegeben empfangen werden. Es kann dann zu Folgefehlern mit Telegrammverlust kommen.
- Protocol Identifier = 0
- Länge im Header ist zwischen 6 und 207
- Der empfangene Funktionscode wird geprüft. Wenn der Funktionscode ungleich 1, 2, 3, 4, 5, 6, 15 oder 16 ist, wird ein Exception Telegramm gesendet.
- Bei Schreibaufträgen müssen die Länge aus dem Header sowie die Register-/Bitanzahl und der Bytecount aus dem Telegramm zusammenpassen.
- Die Register-/Bitanzahl wird überprüft. Bei einer zu großen Anzahl wird ein Exception Telegramm gesendet.
- Bei der Bearbeitung eines Auftrags wird überprüft, ob der durch die Modbusadresse angesprochene Datenbaustein in der erforderlichen Länge vorhanden ist. Die DB-Nummer darf nicht 0 und nicht identisch mit MODBUS_PARAM oder dem zugeordneten Instanz-DB sein. Im Fehlerfall wird ein Exception Telegramm gesendet.

Verbindungsabbau im Fehlerfall

In speziellen Fehlerfällen löst der FB einen Verbindungsabbau aus:

- Überwachungszeit für Verbindungsaufbau abgelaufen
- Überwachungszeit für Empfang abgelaufen
- PI ungleich 0
- empfangener TI ungleich gesendetem (nur bei S7 ist Client)
- Längenangabe im Header passt nicht zur Längenangabe im Telegramm

7.3 Diagnosemeldungen des FB MODBUSPN

Meldungen an den STATUS-Ausgängen des FBs

Am Baustein MODBUSPN gibt es 3 Statusausgänge: STATUS_MODBUS, STATUS_CONN und STATUS_FUNC.
An STATUS_MODBUS werden die Fehlermeldungen bezüglich der Modbustelegrammbearbeitung angezeigt. An STATUS_CONN werden die Fehlermeldungen bezüglich der Verbindungsbearbeitung angezeigt. An STATUS_FUNC wird der Name der Funktion angezeigt, die den Fehler verursacht hat. Die STATUS-Werte sind gültig, wenn **ERROR** gesetzt ist. Nachfolgend finden Sie eine Aufstellung der FB-spezifischen Fehlermeldungen.

Fehlermeldungen der aufgerufenen SFCs und FBs

Die FBs MODBUSPN, MOD_CLI und MOD_SERV benutzen die Standardbausteine SFC6, SFC20, SFC24, SFC51 und SFC52. Die Fehlermeldungen dieser Bausteine werden unverändert an STATUS_MODBUS weitergegeben.

Der in MOD_CLI bzw. MOD_SERV aufgerufene FB TCP_COMM benutzt die Standardbausteine SFB4, FB63, FB64, FB65 und FB66. Die Fehlermeldungen dieser Bausteine werden unverändert an STATUS_CONN weitergegeben.

Im Diagnosepuffer oder der Online-Hilfe zu den SFCs/FBs aus dem SIMATIC Manager finden Sie weitere Hinweise zu diesen Fehlermeldungen.

Fehlermeldungen am Ausgang STATUS_MODBUS		
STATUS (Hex)	**Ereignistext**	**Abhilfe**
A001	Der Parameter-DB MODBUS_PARAM ist zu kurz.	Korrigieren Sie die Länge des DBs MODBUS_PARAM.
A002	Der Parameter *end_x* ist kleiner als *start_x*.	Korrigieren Sie die Angaben im DB MODBUS_PARAM.
A003	Ein DB, auf den MODBUS-Adressen abgebildet werden sollen, ist zu kurz. Minimallänge: - bei Registern: (*end_x* - *start_x* + 1) ∗ 2 - bei Bitwerten: (*end_x* - *start_x* + 7) / 8 weitere mögliche Ursachen: • S7 ist Client: falsche Aufrufparameter • S7 ist Server: falscher Adressbereich im Anforderungstelegramm des Clients	Verlängern Sie den DB. S7 ist Client: Korrigieren Sie die Auftragsparameter START_ ADDRESS oder LENGTH. S7 ist Server: Ändern Sie die Anforderung des Client.
A004	Nur S7 ist Client: Es wurde eine unzulässige Kombination von DATA_TYPE und WRITE_READ angegeben.	Korrigieren Sie die Aufrufparameter. Es können nur die Datentypen 1 und 3 geschrieben werden.

Fehlermeldungen am Ausgang STATUS_MODBUS		
STATUS (Hex)	**Ereignistext**	**Abhilfe**
A005	S7 ist Client: Es wurde ein unzulässiger Wert am Parameter LENGTH angegeben. S7 ist Server: Die Register-/Bitanzahl im Anforderungstelegramm ist unzulässig. Wertebereiche: Coils/Inputs lesen: 1 bis 2000 Coils schreiben: 1 bis 800 Register lesen: 1 bis 125 Holding Register schreiben: 1 bis 100	S7 ist Client: Korrigieren Sie den Parameter LENGTH. S7 ist Server: Verändern Sie im Anforderungstelegramm die Anzahl.
A006	Der über DATA_TYPE, START_ADDRESS und LENGTH angegebene Bereich existiert nicht in *data_type_1* bis *data_type_8*.	S7 ist Client: Korrigieren Sie die Parametrierungskombination DATA_TYPE, START_ ADDRESS und LENGTH. S7 ist Server: Ändern Sie die Anforderung des Client oder korrigieren Sie die Parametrierung an *data_type_x*.
A007	S7 ist Client: Es wurde eine ungültige Überwachungszeit an RECV_TIME oder CONN_TIME parametriert. Für RECV_TIME muss ein Wert >= 20 ms, für CONN_TIME >= 100 ms eingetragen werden.	Korrigieren Sie die Parametrierung.
A009	S7 ist Client: Der empfangene Transaction Identifier TI ist ungleich dem gesendeten. Die Kommunikationsverbindung wird abgebaut.	Überprüfen Sie mit Hilfe einer Telegrammaufzeichnung die Daten des Koppelpartners.
A00A	S7 ist Client: Die empfangene UNIT ist ungleich der gesendeten.	Überprüfen Sie mit Hilfe einer Telegrammaufzeichnung die Daten des Koppelpartners.
A00B	S7 ist Client: Der empfangene Funktionscode ist ungleich dem gesendeten. S7 ist Server: Es wurde ein ungültiger Funktionscode empfangen.	S7 ist Client: Überprüfen Sie mit Hilfe einer Telegrammaufzeichnung die Daten des Koppelpartners. S7 ist Server: Ändern Sie die Anforderung des Client. Der FB MODBUSPN bearbeitet die Funktionscodes 1, 2, 3, 4, 5, 6, 15 und 16.
A00C	Der empfangene Bytecount passt nicht zur Registeranzahl. Die Kommunikationsverbindung wird abgebaut.	Überprüfen Sie mit Hilfe einer Telegrammaufzeichnung die Daten des Koppelpartners.
A00D	Nur bei S7 ist Client: Die Register-/Bitadresse bzw. die Register-/Bitanzahl im Antworttelegramm ist ungleich der im Anforderungstelegramm.	
A00E	Die Längenangabe im modbusspezifischen Telegrammheader passt nicht zu den Angaben der Register-/Bitanzahl oder des Bytecount im Telegramm. Der FB verwirft alle Daten. Die Kommunikationsverbindung wird abgebaut.	

Fehlermeldungen am Ausgang STATUS_MODBUS		
STATUS (Hex)	**Ereignistext**	**Abhilfe**
A00F	Es wurde ein Protocol Identifier ungleich 0 empfangen. Die Kommunikationsverbindung wird abgebaut.	Überprüfen Sie mit Hilfe einer Telegrammaufzeichnung die Daten des Koppelpartners.
A010	Bei den Parametern *db_1* bis *db_8* wurde eine DB-Nummer doppelt vergeben.	Korrigieren Sie die Parametrierung im DB MODBUS_PARAM.
A011	Am Eingangsparameter DATA_TYPE wurde ein unzulässiger Wert angegeben (zulässige Werte sind 1, 2, 3 und 4).	Korrigieren Sie die Aufrufparameter.
A012	Die parametrierten Bereiche *data_type_1* und *data_type*_2 überlappen.	Korrigieren Sie die Parametrierung. Die Datenbereiche dürfen keinen gemeinsamen Registeradressbereich besitzen.
A013	Die parametrierten Bereiche *data_type_1* und *data_type*_3 überlappen.	
A014	Die parametrierten Bereiche *data_type_1* und *data_type_4* überlappen.	
A015	Die parametrierten Bereiche *data_type_1* und *data_type_5* überlappen.	
A016	Die parametrierten Bereiche *data_type_1* und *data_type_6* überlappen.	
A017	Die parametrierten Bereiche *data_type_1* und *data_type_7* überlappen.	
A018	Die parametrierten Bereiche *data_type_1* und *data_type_8* überlappen.	
A019	Einer der Parameter *db_x* wurde auf 0 gesetzt, obwohl der zugehörige *data_type_x* mit > 0 parametriert ist. DB 0 darf nicht verwendet werden, weil dieser für das System reserviert ist. Der Bereich *db_1* ist der Initialbereich und muss parametriert werden.	Korrigieren Sie die Parametrierung am Parameter *db_1* auf > 0 bzw. setzen Sie die verwendeten Parameter *db_x* auf > 0.
A01A	Falsche Länge im Header: Es sind 1 bis 253 Byte zulässig. Die Kommunikationsverbindung wird abgebaut.	Überprüfen Sie mit Hilfe einer Telegrammaufzeichnung die Daten des Koppelpartners.
A01F	Der FB MODBUSPN ist in einem unzulässigen Betriebszustand.	Kontaktieren Sie den Produkt Support.
A023	Die parametrierten Bereiche *data_type*_2 und *data_type_3* überlappen.	Korrigieren Sie die Parametrierung im DB MODBUS_PARAM. Die Datenbereiche dürfen keinen gemeinsamen Registeradressbereich besitzen.
A024	Die parametrierten Bereiche *data_type*_2 und *data_type_4* überlappen.	
A025	Die parametrierten Bereiche *data_type*_2 und *data_type_5* überlappen.	
A026	Die parametrierten Bereiche *data_type*_2 und *data_type_6* überlappen.	
A027	Die parametrierten Bereiche *data_type*_2 und *data_type*_7 überlappen.	
A028	Die parametrierten Bereiche *data_type*_2 und *data_type*_8 überlappen.	

Fehlermeldungen am Ausgang STATUS_MODBUS		
STATUS (Hex)	**Ereignistext**	**Abhilfe**
A034	Die parametrierten Bereiche *data_type_3* und *data_type*_4 überlappen.	Korrigieren Sie die Parametrierung im DB MODBUS_PARAM. Die Datenbereiche dürfen keinen gemeinsamen Registeradressbereich besitzen.
A035	Die parametrierten Bereiche *data_type_3* und *data_type*_5 überlappen.	
A036	Die parametrierten Bereiche *data_type_3* und *data_type_6* überlappen.	
A037	Die parametrierten Bereiche *data_type_3* und *data_type*_7 überlappen.	
A038	Die parametrierten Bereiche *data_type_3* und *data_type*_8 überlappen.	
A045	Die parametrierten Bereiche *data_type_4* und *data_type*_5 überlappen.	
A046	Die parametrierten Bereiche *data_type_4* und *data_type_6* überlappen.	
A047	Die parametrierten Bereiche *data_type_4* und *data_type*_7 überlappen.	
A048	Die parametrierten Bereiche *data_type_4* und *data_type*_8 überlappen.	
A056	Die parametrierten Bereiche *data_type_5* und *data_type_6* überlappen.	
A057	Die parametrierten Bereiche *data_type_5* und *data_type_7* überlappen.	
A058	Die parametrierten Bereiche *data_type_5* und *data_type_8* überlappen.	
A067	Die parametrierten Bereiche *data_type_6* und *data_type_7* überlappen.	
A068	Die parametrierten Bereiche *data_type_6* und *data_type_8* überlappen.	
A078	Die parametrierten Bereiche *data_type_7* und *data_type_8* überlappen.	
A079	Die am Parameter ID angegebene Verbindungs-ID ist im Parameter-DB MODBUS_PARAM nicht enthalten.	Korrigieren Sie die Parametrierung am Eingang ID.
A07A	Es wurde ein unzulässiger Wert am Parameter ID angegeben (Wertebereich von 1 bis 4095).	
A07B	Die angegebene ID ist im Parameter-DB doppelt enthalten.	Korrigieren Sie die Parametrierung im DB MODBUS_PARAM.
A07C	Im Parameter-DB wurde ein unzulässiger Wert am Parameter *data_type_x* angegeben (zulässige Werte sind 0 bis 4).	
A07D	Im Parameter-DB enthält der Parameter *data_type_1* keinen Eintrag. Der Parameterbereich „*_1*“ ist der Initialbereich und muss parametriert werden.	
A07E	An db_x wurde die Nummer des Parameter-DBs MODBUS_PARAM oder die Nummer des Instanz-DBs vom Baustein MODBUSPN angegeben.	

Fehlermeldungen am Ausgang STATUS_MODBUS		
STATUS (Hex)	**Ereignistext**	**Abhilfe**
A07F	Der an DB_PARAM angegebene DB ist kein Modbus-Parameter-DB. Es wurde die Längeninformation im DBW0 geändert oder ein falscher DB angegeben.	Korrigieren Sie die Parametrierung am Eingang DB_PARAM.
A081	Nur bei S7 ist Client und Funktionscode 5: Die Daten des Antworttelegramms sind nicht das Echo der Anforderung.	Überprüfen Sie mit Hilfe einer Telegrammaufzeichnung die Daten des Koppelpartners.
A082	Nur bei S7 ist Client und Funktionscode 6: Der empfangene Registerwert ist ungleich dem gesendeten.	Überprüfen Sie mit Hilfe einer Telegrammaufzeichnung die Daten des Koppelpartners.
A083	S7 ist Client: Es ist ein Auftrag angestoßen worden, während der vorherige Auftrag noch läuft. Der Auftrag wird nicht ausgeführt.	Stossen Sie erst dann einen neuen Auftrag an, wenn der vorherige Auftrag mit DONE_ NDR = TRUE oder ERROR = TRUE beendet wurde.
A084	Es konnte keine Identifizierungskennung IDENT_CODE für die Lizenzierung ermittelt werden.	Wenden Sie sich an den Produkt Support.
A090	Der Baustein ist für diese CPU noch nicht lizenziert.	Lesen Sie die Identifikations-kennung für diese CPU am Ausgang IDENT_CODE aus und fordern Sie damit den Freischaltungscode bei IT4Industry an. Siehe Kapitel 5 „Lizenzierung“.
A091	Als Antwort wurde ein Exception Telegramm mit Exception Code 1 empfangen (nur bei CP ist Client).	Der Koppelpartner unterstützt die angeforderte Funktion nicht.
A092	Als Antwort wurde ein Exception Telegramm mit Exception Code 2 empfangen (nur bei CP ist Client). Es erfolgte ein Zugriff auf eine nicht vorhandene/ nicht zulässige Adresse beim Koppelpartner.	Korrigieren Sie LENGTH bzw. START_ADDRESS beim FB-Aufruf.
A093	Als Antwort wurde ein Exception Telegramm mit Exception Code 3 empfangen (nur bei CP ist Client).	Überprüfen Sie die Fehleranzeigen beim Koppelpartner.
A094	Als Antwort wurde ein Exception Telegramm mit Exception Code 4 empfangen (nur bei CP ist Client).	Überprüfen Sie die Fehleranzeigen beim Koppelpartner.
A095	Als Antwort wurde ein Exception Telegramm mit einem unbekannten Exception Code empfangen (nur bei CP ist Client)	Kontrollieren Sie die Fehler-meldungen des Koppel-partners und überprüfen Sie ggf. die Daten mit einer Telegrammaufzeichnung.

Fehlermeldungen am Ausgang STATUS_CONN		
STATUS (Hex)	**Ereignistext**	**Abhilfe**
A100	Für einen Auftrag ist die Überwachungszeit CONN_TIME oder RECV_TIME abgelaufen. Bei Ablauf der RECV_TIME wird die Verbindung abgebaut.	Prüfen Sie die Parametrierung der Verbindung.
A101	Die interne Überwachungszeit der Funktion TDISCON ist abgelaufen.	Kontaktieren Sie den Produkt Support.

7.4 Diagnosemeldungen der verwendeten Bausteine

Fehlermeldungen der Bausteine SFC 6, SFC 20, SFC51 und SFC52 am Ausgang STATUS_MODBUS		
STATUS (Hex)	**Ereignistext**	**Abhilfe**
7xxx	Bitte entnehmen Sie die detaillierte Information der Online-Hilfe des SIMATIC Manager.	Siehe Online-Hilfe (SIMATIC Manager -> Baustein markieren -> Taste F1)
8xxx	Bitte entnehmen Sie die detaillierte Information der Online-Hilfe des SIMATIC Manager.	Siehe Online-Hilfe (SIMATIC Manager -> Baustein markieren -> Taste F1)

Fehlermeldungen der Bausteine FB 63, FB 64, FB 65 und FB 66 am Ausgang STATUS_CONN		
STATUS (Hex)	**Ereignistext**	**Abhilfe**
7xxx	Bitte entnehmen Sie die detaillierte Information der Online-Hilfe des SIMATIC Manager.	Siehe Online-Hilfe (SIMATIC Manager -> Baustein markieren -> Taste F1)
8xxx	Bitte entnehmen Sie die detaillierte Information der Online-Hilfe des SIMATIC Manager.	Siehe Online-Hilfe (SIMATIC Manager -> Baustein markieren -> Taste F1)

7.5 Diagnosemeldungen des SFC24

Fehlermeldungen des SFC24 am Ausgang STATUS_MODBUS		
STATUS (Hex)	**Ereignistext**	**Abhilfe**
80A1	DB Nummer = 0 oder zu groß für die CPU Diese Fehlermeldung kann auch auftreten, wenn für den Aufruf des FB MODBUSPN im OB1 bzw. Weckalarm-OB und OB100 unterschiedliche Instanz-DBs verwendet werden.	Wählen Sie eine zulässige DB Nummer. Verwenden Sie für den MODBUSPN-Aufruf im OB1 bzw. Weckalarm-OB und im OB100 denselben IDB.
80B1	Der DB existiert nicht auf der CPU.	Alle Datenbausteine, die an *db_x* angegeben werden, müssen angelegt und auf die CPU übertragen werden.
80B2	DB UNLINKED	DB nicht als UNLINKED generieren.

8 Applikationsbeispiel

Allgemein

Im Folgenden ist die Verwendung des MODBUSPN FB in einem einfachen Programmbeispiel dargestellt.

Bitte beachten Sie, dass das mitgelieferte Beispielprojekt nur eine Orientierungshilfe für den Umgang mit den Modbus-Bausteinen darstellt und nicht für den Einsatz in Produktiv-Anlagen geeignet ist.

Beispiel-Projekt auf der CD

Auf der CD finden Sie ein umfangreiches Beispielprojekt in dem für alle Funktionsvarianten SIMATIC-Stationen angelegt wurden.

- SIMATIC-Station ist S7-300, S7-400 oder IM 151-8 PN/DP CPU
- SIMATIC-Station ist Client oder Server

Programmbeispiel

Das Programmbeispiel besteht aus den Bausteinen:

- Anlauf-Baustein OB100 mit Aufruf des FB102
- Programmierfehler-OB121 mit Aufruf des FB102
- Zyklischer Betrieb OB1 mit Aufruf des FB102
- Globale Datenbausteine zum Auftragsanstoß (z.B. mit Hilfe einer Variablentabelle) und für die Lizenzierung
- Datenbausteine für Register- und Bitwerte

Verwendete Bausteine

Die aufgelisteten Bausteine werden im mitgelieferten Beispielprojekt für S7-Stationen mit FB MODBUSPN verwendet.

Baustein	Symbol	Kommentar
OB 1	CYCL_EXC	Zyklische Programmbearbeitung
OB 100	COMPLETE RESTART	Anlauf-OB für Neustart
OB 121	PROG_ERR	Programmierfehler-OB
FB 102	MODBUSPN	FB MODBUSPN
FB 103	TCP_COMM	FB TCP_COMM
FB 104	MOD_CLI	FB MOD_CLI
FB 105	MOD_SERV	FB MOD_SERV
DB 1	CONTROL_DAT	Arbeits-DB CONTROL DAT für FB MODBUSPN
DB 2	MODBUS_PARAM	Parameter-DB PARAM_DB für FB MODBUSPN
DB 3	LICENSE_DB	Lizenz-DB für FB MODBUSPN
DB 102	IDB_MODBUS	Instanz-DB für FB MODBUSPN
DB 11	DATA_AREA_1	Werte-DB für Bereich 1
DB 12	DATA_AREA_2	Werte-DB für Bereich 2
DB 13	DATA_AREA_3	Werte-DB für Bereich 3
DB 14	DATA_AREA_4	Werte DB für Bereich 4
DB 15	DATA_AREA_5	Werte-DB für Bereich 5

A Literatur

MODBUS IDA MODBUS APPLICATION PROTOCOL SPECIFICATION
V1.1b, December 28, 2006

http://www.modbus-IDA.org

Glossar

A

Adresse Die Adresse gibt den physikalischen Speicherplatz an und ermöglicht den direkten Zugriff auf den Operanden, der unter dieser Adresse abgespeichert ist.

ANLAUF Der Betriebszustand ANLAUF wird beim Übergang vom Betriebszustand STOP in den Betriebszustand RUN durchlaufen.

Anwenderprogramm Das Anwenderprogramm enthält alle Anweisungen und Vereinbarungen für die Signalverarbeitung, durch die eine Anlage oder ein Prozess gesteuert werden können. Das Anwenderprogramm wird bei SIMATIC S7 strukturiert aufgebaut und in kleinere Einheiten, die Bausteine, unterteilt.

Arbeitsspeicher Der Arbeitsspeicher ist ein RAM-Speicher in der CPU, auf den der Prozessor während der Bearbeitung des Anwenderprogramms zugreift.

Automatisierungs-system Ein Automatisierungssystem ist eine speicherprogrammierbare Steuerung, die aus mindestens einer Zentralbaugruppe, verschiedenen Ein- und Ausgabebaugruppen sowie Bedien- und Beobachtungsgeräten besteht.

B

Baudrate → Übertragungsgeschwindigkeit

Baugruppe Baugruppen sind steckbare Leiterplatten für Automatisierungssysteme.

Baugruppenparameter Baugruppenparameter sind Werte, mit denen das Verhalten der Baugruppe eingestellt werden kann. Man unterscheidet zwischen statischen und dynamischen Baugruppenparametern.

Baustein Bausteine sind durch ihre Funktion, ihre Struktur oder ihren Verwendungszweck abgegrenzte Teile des Anwenderprogramms. Es gibt bei STEP 7

- Codebausteine (FB, FC, OB, SFB, SFC)
- Datenbausteine (DB, SDB) und
- Anwenderdefinierte Datentypen (UDT)

Bausteinaufruf Als Bausteinaufruf bezeichnet man die Verzweigung der Programmbearbeitung in den aufgerufenen Baustein.

Bausteinparameter Bausteinparameter sind Platzhalter innerhalb mehrfach nutzbarer Bausteine, die beim Aufruf des betreffenden Bausteins mit aktuellen Werten versorgt werden.

Betriebssystem der CPU Das Betriebssystem der CPU organisiert alle Funktionen und Abläufe der CPU, die nicht mit einer speziellen Steuerungsaufgabe verbunden sind.

Bussegment Teil eines -> Subnetzes. Subnetze können aus Bussegmenten mittels Segmentübergängen wie Repeater und Bridges gebildet sein. Segmente sind für die Adressierung transparent.

Betriebszustand Die Automatisierungsfamilie SIMATIC S7 kennt drei unterschiedliche Betriebszustände: STOP, ANLAUF und RUN. Die Funktionalität der CPUs in den einzelnen Betriebszuständen ist unterschiedlich.

C

Client Unter Client wird ein Gerät, oder allgemein ein Objekt verstanden, das einen -> Server auffordert, einen Dienst zu erbringen.

CPU Central Processing Unit = Zentralbaugruppe des S7-Automatisierungssystems mit Steuer- und Rechenwerk, Speicher, Systemprogramm und Schnittstellen zu Peripheriebaugruppen.

D

Datenbaustein (DB) Datenbausteine sind Bausteine, die Daten und Parameter enthalten, mit denen das Anwenderprogramm arbeitet. Sie enthalten im Gegensatz zu allen anderen Bausteinen keine Anweisungen. Es gibt globale Datenbausteine und Instanzdatenbausteine.
Auf die in den Datenbausteinen enthaltenen Daten kann absolut oder symbolisch zugegriffen werden. Komplexe Daten können strukturiert abgelegt werden.

Datentyp Mit Hilfe der Datentypen können Sie festlegen, wie der Wert einer Variablen oder Konstanten im Anwenderprogramm verwendet werden soll. Die Datentypen sind unterteilt in elementare und strukturierte Datentypen.

Default-Einstellung Die Default-Einstellung ist eine sinnvolle Grundeinstellung, die immer dann verwendet wird, wenn kein anderer Wert eingegeben wird.

Diagnoseereignisse Diagnoseereignisse sind z. B. Fehler auf einer Baugruppe, Systemfehler in der CPU, die z. B. durch einen Programmfehler hervorgerufen wurden oder Übergänge von Betriebszuständen.

Diagnosefunktionen Die Diagnosefunktionen umfassen die gesamte Systemdiagnose und beinhalten Erkennen, Auswerten und Melden von Fehlern innerhalb des Automatisierungssystems.

Diagnosepuffer Jede CPU hat einen Diagnosepuffer, in den zu allen Diagnoseereignissen in der Reihenfolge ihres Auftretens nähere Informationen eingetragen werden.

F

Funktionsbausteine (FBs)
Funktionsbausteine sind Bestandteile des Anwenderprogramms und sind gemäß der IEC-Norm ”Bausteine mit Gedächtnis”. Das Gedächtnis für den Funktionsbaustein ist ein zugeordneter Datenbaustein, der ”Instanz-Datenbaustein”. Funktionsbausteine sind parametrierbar. Sie können mit und ohne Parameter genutzt werden.

H

Hardware
Als Hardware bezeichnet man die gesamte physikalische und technische Ausstattung eines Automatisierungssystems.

I

Industrial Ethernet
Ein Bussystem nach IEEE 802.3 (ISO 8802-2)

Instanz-Datenbaustein
Ein Instanz-Datenbaustein ist ein einem Funktionsbaustein zugeordneter Baustein, der Daten für diesen speziellen Funktionsbaustein enthält.

Interrupt
Interrupt ist eine Bezeichnung für die Unterbrechung der Programmbearbeitung im Prozessor eines Automatisierungssystems durch einen von außen anstehenden Alarm.

K

Konfigurierung
Unter Konfigurierung versteht man die Zusammenstellung einzelner Baugruppen eines Automatisierungssystems.

L

Laden aus PG
Laden von Ladeobjekten (z. B. Codebausteine) vom Programmiergerät in den Ladespeicher der Zentralbaugruppe (CPU).

Laden in PG
Laden von Ladeobjekten (z. B. Codebausteine) aus dem Ladespeicher der Zentralbaugruppe in das Programmiergerät.

M

MAC-Adresse
Adresse zur Unterscheidung von verschiedenen Stationen, die an einem gemeinsamen Übertragungsmedium (Industrial Ethernet) angeschlossen sind.

Media Access Control (MAC)
Steuerung des Zugriffs einer Station auf ein, gemeinsam mit anderen Stationen benutztes, Übertragungsmedium.

N

NCM S7 für Industrial Ethernet — Projektiersoftware zur Projektierung und Diagnose von Ethernet-CP.

O

Online/Offline — Bei Online besteht eine Datenverbindung zwischen Automatisierungssystem und Programmiergerät, bei Offline nicht.

Online-Hilfe — STEP 7 bietet Ihnen die Möglichkeit, sich während des Arbeitens mit der Programmiersoftware kontextabhängige Hilfetexte am Bildschirm anzeigen zu lassen.

Operand — Ein Operand ist Teil einer STEP 7-Anweisung und sagt aus, womit der Prozessor etwas tun soll. Er kann sowohl absolut als auch symbolisch adressiert werden.

P

Parameter — Parameter sind Werte, die man vergeben kann. Man unterscheidet zwischen Bausteinparametern und Baugruppenparametern.

Parametrierung — Unter Parametrierung versteht man das Einstellen des Verhaltens einer Baugruppe.

Protokoll — Die Kommunikationspartner einer Datenübertragung müssen sich an feste Regeln für die Abwicklung und Durchführung des Datenverkehrs halten. Diese Regeln werden Protokolle genannt.

Prozedur — Der Ablauf einer Datenübertragung nach einem bestimmten Protokoll wird als Prozedur bezeichnet.

Prozessabbild — Das Prozessabbild ist ein besonderer Speicherbereich im Automatisierungssystem. Am Anfang des zyklischen Programms werden die Signalzustände der Eingabebaugruppen zum Prozessabbild der Eingänge übertragen. Am Ende des zyklischen Programms wird das Prozessabbild der Ausgänge als Signalzustand zu den Ausgabebaugruppen übertragen.

R

Rack — Ein Rack ist ein Baugruppenträger, der Steckplätze für Baugruppen enthält.

S

Schnittstellenmodul Auf dem Schnittstellenmodul erfolgt die physikalische Umsetzung von Signalen. Durch Austausch der steckbaren Schnittstellenmodule können Sie den Kommunikationsprozessor an die Physik des Kommunikationspartners anpassen.

Server Ein Server ist ein Gerät, oder allgemein ein Objekt, das bestimmte Dienste erbringen kann; aufgrund der Anforderung durch einen -> Client wird der Dienst erbracht.

Software Die Gesamtheit aller Programme, die auf einem Rechensystem eingesetzt werden, bezeichnet man als Software. Dazu gehören das Betriebssystem und die Anwenderprogramme.

STEP7 STEP7 ist die Programmiersoftware von SIMATIC S7.

Subnetz Ein Subnetz ist ein Teil eines -> Netzes, dessen Parameter abgeglichen werden müssen. Es umfasst die Buskomponenten und alle angeschlossenen Stationen. Subnetze können beispielsweise mittels -> Gateways zu einem Netz gekoppelt werden.
Eine Anlage besteht aus mehreren Subnetzen mit eindeutigen Subnetznummern. Ein Subnetz besteht aus mehreren ->Teilnehmern mit eindeutigen -> MAC-Adressen (bei Industrial Ethernet).

Systembausteine Systembausteine unterscheiden sich von den anderen Bausteinen dadurch, dass sie im System S7-400 bereits integriert sind und für bereits definierte Systemfunktionen zur Verfügung stehen. Es gibt Systemdatenbausteine, Systemfunktionen und Systemfunktionsbausteine.

Systemfunktionen (SFCs) Systemfunktionen sind Bausteine ohne Gedächtnis, die bereits im Betriebssystem der CPU integriert sind und bei Bedarf vom Anwender aufgerufen werden können.

Systemfunktions-bausteine (SFBs) Systemfunktionsbausteine sind Bausteine mit Gedächtnis, die bereits im Betriebssystem der CPU integriert sind und bei Bedarf vom Anwender aufgerufen werden können.

T

Tool Ein Tool ist ein Software-Werkzeug, das auf Funktionen des Betriebssystems im PG/PC zugreifen kann.

Teilnehmer Ein Teilnehmer wird durch eine -> MAC-Adresse an -> Industrial Ethernet identifiziert.

U

Übertragungs-geschwindigkeit Ist nach DIN 44302 die Anzahl der je Zeiteinheit übertragenen Binär-entscheidungen. Die Einheit ist bit/sec. Die Wahl der Übertragungs-geschwindigkeit hängt von verschiedenen Randbedingungen, wie beispielsweise der Entfernung ab.

V

Variable

Eine Variable ist ein Operand (z. B. E 1.0), der einen symbolischen Namen haben kann und deshalb auch symbolisch adressiert werden kann.

Verbindungs-projektierung

Verbindungsprojektierung bezeichnet die Angabe einer Verbindungs-ID im Systemfunktionsbaustein. Über die Verbindungs-ID können die Systemfunktionsbausteine zwischen zwei Kommunikationsendpunkten kommunizieren.

Z

Zyklische Programm-bearbeitung

Bei der zyklischen Programmbearbeitung läuft das Anwenderprogramm in einer sich ständig wiederholenden Programmschleife ab, die Zyklus genannt wird.

Zykluszeit

Die Zykluszeit ist die Zeit, die die CPU für eine einmalige Bearbeitung des Anwenderprogramms benötigt.

Customer Support

Siemens AG
Industry Sector
I IS IN E&C
Werner-von-Siemens-Str. 60
91052 Erlangen
Tel: (0 91 31) 7-46111
Fax: (0 91 31) 7-44757
Mail:it4.industry@siemens.com

http://www.siemens.com/s7modbus

Siemens Aktiengesellschaft Änderungen vorbehalten Stand: 06/2009